

**T.C.
SÜLEYMAN DEMİREL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**WEB ORTAMINDAKİ YAZILI SINAVLARIN DOĞAL DİL İŞLEME
YÖNTEMLERİ İLE DEĞERLENDİRİLMESİ**

Ebru YILMAZ İNCE

**Danışman
Prof. Dr. Akif KUTLU**

**DOKTORA TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
ISPARTA - 2016**



© 2016 [Ebru YILMAZ İNCE]

TEZ ONAYI

Ebru YILMAZ İNCE tarafından hazırlanan "Web Ortamındaki Yazılı Sınavların Doğal Dil İşleme Yöntemleri ile Değerlendirilmesi" adlı tez çalışması aşağıdaki jüri üyeleri önünde Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda DOKTORA TEZİ olarak başarı ile savunulmuştur.

Danışman

Prof. Dr. Akif KUTLU
Süleyman Demirel Üniversitesi



Jüri Üyesi

Doç. Dr. Bayram CETİŞLİ
Süleyman Demirel Üniversitesi



Jüri Üyesi

Doç. Dr. Abdülkadir ÇAKIR
Süleyman Demirel Üniversitesi



Jüri Üyesi

Doç. Dr. Ahmet Turan ÖZCERİT
Sakarya Üniversitesi



Jüri Üyesi

Yrd. Doç. Dr. Muhammer İLKUÇAR
Mehmet Akif Üniversitesi



Enstitü Müdürü

Doç.Dr.Yasin TUNCER

TAAHHÜTNAME

Bu tezin akademik ve etik kurallara uygun olarak yazıldığını ve kullanılan tüm literatür bilgilerinin referans gösterilerek tezde yer aldığını beyan ederim.

Ebru YILMAZ İNCE



İÇİNDEKİLER

	Sayfa
İÇİNDEKİLER.....	i
ÖZET	iii
ABSTRACT	iv
TEŞEKKÜR.....	v
ŞEKİLLER DİZİNİ	vi
ÇİZELGELER DİZİNİ	vii
SİMGELER VE KISALTMALAR DİZİNİ.....	viii
1. GİRİŞ.....	1
1.1. Tezin Amacı ve Motivasyonu	5
1.2. Tez Konusunun Önemi	5
2. KAYNAK ÖZETLERİ.....	6
2.1. Türkçe Doğal Dil İşleme Çalışmaları	9
2.1.1. Biçimbilimsel analiz çalışmaları	10
2.1.2. Sözdizilimsel analiz çalışmaları	12
2.1.3. Anlamsal analiz çalışmaları	14
2.1.4. Problem analizi çalışmaları	16
2.2. Otomatik Sınav Puanlama Sistemi.....	19
2.3. Metin Benzerliği	25
2.3.1. Sözcüksel metin benzerliği	25
2.3.2. Anlamsal metin benzerliği	25
3. YÖNTEM	27
3.1. Otomatik Türkçe Sınav Puanlama Sistemi Yazılımının Tasarlanması	27
3.2. Geliştirilen TSPS Yazılımının Hesaplama Modülü.....	32
3.3. TGAA ile Gizli Anlambilimsel Çekirdek Yönteminin Karşılaştırılması	43
3.4. TSPS Yazılımının Değerlendirme Yöntemi	46
4. ARAŞTIRMA BULGULARI.....	50
4.1. TSPS Yazılımının Kullanıcı Arayüzleri	50
4.2. TSPS Yazılımının Yeterlilik Testi	52
5. SONUÇ VE ÖNERİLER.....	62
KAYNAKLAR	64
EKLER.....	76
EK A. Metin Araçları Sınıfı.....	77
EK B. Türkçe Gereksiz Kelimeler	79
EK C. Kök Bulucu	80
EK D. Eş Anlam Bulucu.....	81
EK E. Yakın Anlam Bulucu	83
EK F. Kosinüs Benzerliği.....	85
EK G. Kısaltma Bulucu Sınıfı.....	87
EK H. Rakamların Yazıya Çevrilmesi	88
EK I. GAA Fonksiyonu.....	90
EK İ. TGAA Fonksiyonu	93
EK J. Bilgisayar Ağları Deneme Sınavı.....	98
EK K. Hata Denetimi Kodları.....	99
EK L. Levestein Düzenleme Uzaklık Algoritması	101

EK M. Birinci Cevap Anahtarına Göre Kosinüs Benzerlik, GAA ve TGAA Puanları	102
EK N. İkinci Cevap Anahtarına Göre Kosinüs Benzerlik, GAA ve TGAA Puanları	105
EK O. Eğitimci1 Cevap Anahtarı	108
EK Ö. Eğitimci2 Cevap Anahtarı	110
EK P. TSPS Puan ile Eğitimci1 Puan Pearson t Testi Sonuçları	112
EK R. TSPS Puan ile Eğitimci2 Puan Pearson t Testi Sonuçları	113
EK S. TSPSORT Puan ile Eğitimci1 Puan Pearson t Testi Sonuçları	114
EK Ş. TSPSORT Puan ile Eğitimci2 Puan Pearson t Testi Sonuçları	115
ÖZGEÇMİŞ	116



ÖZET

Doktora Tezi

WEB ORTAMINDAKİ YAZILI SINAVLARIN DOĞAL DİL İŞLEME YÖNTEMLERİ İLE DEĞERLENDİRİLMESİ

Ebru YILMAZ İNCE

Süleyman Demirel Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Akif KUTLU

Otomatik sınav puanlama sistemleri, öğrencilerin sınav cevaplarına bilgisayar yardımıyla otomatik bir şekilde puan atamasını sağlamaktadır. Otomatik sınav puanlama sistemleri, kişilerin eğitildiği dile uygun olmalıdır. Farklı dillerde geliştirilmiş birçok otomatik sınav puanlama sistemi vardır ve bunların çoğu İngilizce diline uygun geliştirilmiştir.

Bu tez çalışmasında, sınav sorularını depolamak ve çevrimiçi kısa cevaplı sorularla hazırlanmış sınavları yapmak ve otomatik olarak puanlamak için, web tabanlı otomatik Türkçe sınav puanlama sistemi (TSPS) yazılımı geliştirilmiştir. TSPS yazılımında, anlamsal ilişkiler için Türkçe WordNet ve doğal dil işleme alt yapısı için Zemberek kullanılmıştır. TSPS yazılımı kosinüs benzerliği ve tümleşik gizli anlambilimsel analiz (TGAA) metotlarını içeren hibrid bir model kullanmaktadır. TSPS yazılımının doğruluğunu ölçmek için, Süleyman Demirel Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü Uzaktan Eğitim Programında bir özel durum çalışması gerçekleştirilmiştir. Çalışmanın sonuçlarına göre, %92'lik başarı oranı ile TSPS yazılımının Türkçe otomatik sınav puanlaması için kullanılabilir olduğu tespit edilmiştir.

Anahtar Kelimeler: Otomatik sınav puanlama, doğal dil işleme, gizli anlambilimsel analiz, kosinüs benzerliği, tümleşik gizli anlambilimsel analiz.

2016, 117 sayfa

ABSTRACT

Ph.D. Thesis

ASSESSMENT OF WRITTEN EXAMS IN WEB ENVIRONMENT WITH NATURAL LANGUAGE PROCESSING METHODS

Ebru YILMAZ İNCE

**Süleyman Demirel University
Graduate School of Applied and Natural Sciences
Department of Computer Engineering**

Supervisor: Prof. Dr. Akif KUTLU

Automated essay scoring systems calculates a score automatically according to answers of students by the help of the computers. Automated essay scoring system must be appropriate to the language through which people are educated. There are several automated essay scoring systems in different languages and most of them have been developed for the English language.

In this thesis, web-based automated Turkish Essay Scoring System (TSPS) software has been developed to store exam questions and to make online exams for scoring short-answer question essays automatically. In the TSPS software, a Turkish WordNet has been used for semantic relations and Zemberek has been used as a generic natural language processing framework. The TSPS software uses a hybrid model of Cosine Similarity and Integrated Latent Semantic Analysis (TGAA) methods. In order to measure of the TSPS software accuracy, a case study has been performed in Suleyman Demirel University Faculty of Engineering Department of Computer Engineering Distance Education Programme. The results showed that the TSPS software can be used for automated essay scoring in Turkish with a 92% success rate.

Keywords: Automated essay scoring, natural language processing, latent semantic analysis, cosine similarity, integrated latent semantic analysis.

2016, 117 pages

TEŞEKKÜR

Bu tez çalışması boyunca, desteęi ve güveniyle bana başarı azmi aşılayan, beni ilgi alanlarım doęrultusunda yönlendiren deęerli danıřman hocam Prof.Dr. Akif KUTLU'ya teőekkür ederim.

Tezimin gerekleřmesinde 114E952 numaralı proje ile maddi destek saęlayan Türkiye Bilimsel ve Teknolojik Arařtırma Kurumu'na (TÜBİTAK'a) teőekkür ederim.

2211-Yurt İi Doktora Burs Programı kapsamında saęladığı destekten ötürü TÜBİTAK Bilim İnsani Destekleme Daire Başkanlığı birimine teőekkür ederim.

Balkanet projesinde geliřtirilen Türke Wordnet içeriğini, tez çalışmamda kullanmama izin veren Prof.Dr. Kemal OFLAZER'e teőekkür ederim.

Tez kapsamında gerekleřtirilen 114E952 numaralı Türkiye Bilimsel ve Teknolojik Arařtırma Kurumu (TÜBİTAK) projesinde arařtırmacı olarak alıřan ve tezime katkı saęlayan Do.Dr. Bayram CETİŐLİ'ye ve Do.Dr. Mustafa KO'a teőekkür ederim.

Beni yetiřtiren ve iyi bir eęitim almamı saęlayan annem Ayőe YILMAZ ve babam Arif YILMAZ'a teőekkür ederim.

Her konuda destek olan eőim Murat İNCE'ye teőekkür ederim.

Ebru YILMAZ İNCE
ISPARTA, 2016

ŞEKİLLER DİZİNİ

	Sayfa
Şekil 2.1. Dilin matematiksel analizi.....	7
Şekil 2.2. Dilin morfolojik analizi.....	7
Şekil 3.1. TSPS yazılım mimarisi	28
Şekil 3.2. Zemberek kök ağacı.....	29
Şekil 3.3. TSPS yazılım veri tabanı	32
Şekil 3.4. Hesaplama yönteminin belirlenmesi	33
Şekil 3.5. Kosinüs benzerliği vektörel gösterimi	34
Şekil 3.6. TSPS kosinüs benzerliği hesaplaması.....	35
Şekil 3.7. Tekil değer ayrışımı.....	36
Şekil 3.8. Gizli anlambilimsel analiz terim/doküman matrisi	37
Şekil 3.9. Tekil değer ayrışımı sonuçları.....	37
Şekil 3.10. Örnek GAA kelime vektörü.....	39
Şekil 3.11. Örnek TGAA kelime vektörü.....	39
Şekil 3.12. Eş anlam bulucu algoritması	40
Şekil 3.13. Yakın anlam bulucu algoritması.....	42
Şekil 3.14. Türkçe WordNet düğümleri.....	42
Şekil 3.15. TGAA hesaplaması.....	43
Şekil 3.16. Eğitimci, TGAA ve gizli anlambilimsel çekirdek puanları	45
Şekil 3.17. TSPS analiz ve puanlama aşamaları	47
Şekil 4.1. TSPS yazılımı giriş ekranı.....	50
Şekil 4.2. Eğitimci için sınav hazırlama ekran görüntüsü.....	51
Şekil 4.3. Eğitimci için otomatik puanlama sonuçları ekran görüntüsü	51
Şekil 4.4. Öğrenci için otomatik sınav puanlama sistemi sonucu ekran görüntüsü.....	52
Şekil 4.5. Puanlamalar.....	53
Şekil 4.6. Eğitimci1 ve TSPS puanları karşılaştırması	54
Şekil 4.7. Eğitimci2 ve TSPS puanları karşılaştırması	56
Şekil 4.8. Eğitimci1, Eğitimci2 ve TSPSORT puanları karşılaştırması.....	58

ÇİZELGELER DİZİNİ

	Sayfa
Çizelge 2.1. Otomatik sınav puanlama sistemlerinin karşılaştırılması.....	22
Çizelge 3.1. Türkçe WordNet bileşenleri.....	31
Çizelge 3.2. Eğitimci, TGAA ve gizli anlambilimsel çekirdek puanları.....	44
Çizelge 3.3. Puanlararası Pearson korelasyonu sonuçları.....	45
Çizelge 4.1. Eğitimci1 ve TSPS puanları.....	55
Çizelge 4.2. Eğitimci2 ve TSPS puanları.....	57
Çizelge 4.3. Eğitimci1 ve TSPSORT puanları.....	58
Çizelge 4.4. Eğitimci2 ve TSPSORT puanları.....	59
Çizelge 4.5. Eğitimci1 ve TSPSORT puanları t testi sonuçları.....	60
Çizelge 4.6. Eğitimci2 ve TSPSORT puanları t testi sonuçları.....	61



SİMGELER VE KISALTMALAR DİZİNİ

AES	Automated essay scoring
AWA	Admissions test analytical writing assessment
BETSY	Bayesian essay test scoring system
CLESA	Cross-language explicit semantic analysis
DISCO	Extracting distributionally similar words using cooccurrences
ESA	Explicit semantic analysis
GMAT	Graduate management admissions test
GAA	Gizli anlambilimsel analiz
IEA	Intelligent essay assessor
LCS	Longest common substring
LSA	Latent semantic analysis
LSD	Latent semantical dimensions
NGD	Normalized google distance
OSI	Open systems interconnection
ÖSYM	Ölçme, seçme ve yerleştirme merkezi
PEG	Project essay grade
PMI-IR	Pointwise mutual information-information retrieval
TGAA	Tümleşik gizli anlambilimsel analiz
TSPS	Türkçe sınav puanlama sistemi
TWE	Test of written english
μ	Yakın anlamlılık katsayısı
n	Kelime sayısı

1. GİRİŞ

Uzaktan eğitim, bireylere kendi kendilerine öğrenme imkânı sağlayan, geleneksel eğitimlere göre daha esnek ve birey koşullarına uyarlanabilir bir eğitimdir (İşman, 2008). Uzaktan eğitimin amacı, bireyin özel durumları göz önünde bulundurularak yüz yüze eğitim alma imkânına sahip olmayanların ihtiyacını karşılamaktır (Beldarrain, 2006). Uzaktan eğitim, on sekizinci yüzyılın ortalarında posta yoluyla yapılmaktayken (Moore, 1993) günümüzde kullanılan teknolojik imkânlar sayesinde İnternette web tabanlı olarak gerçekleşmektedir. Web tabanlı öğretim, İnternet kaynaklarını kullanan hipermedya tabanlı, öğrenimin teşvik edildiği ve desteklendiği anlamlı bir öğretim ortamıdır (Bay ve Tüzün, 2002). Bireyselleştirilmiş öğretim, öğrenciler arasındaki öğrenme ve bireysel farklılıkların giderilerek her öğrenci hızına uygun öğretim yapılması tekniğidir. Web tabanlı eğitim, bireysel öğretim tekniğine uygun bir sistemdir (Hatzilygeroudis vd., 2004). Öğrenme yönetim sistemleri; uzaktan ve web tabanlı eğitimde öğrenenlerin ders seçimi ve derse kayıt olmasına, ders içeriklerinin sunulmasına, ölçme ve değerlendirme yapılmasına, kullanıcı bilgilerinin izlenip raporlanmasına olanak sağlayan bir yönetim yazılımıdır. Günümüzde teknolojik gelişmeler sayesinde, öğrenme yönetim sistemlerinin daha verimli olması ve öğrenme yönetim sistemlerinde açık uçlu sınavların değerlendirilmesi için otomatik sınav puanlandırma sistemleri geliştirilmiştir.

Otomatik sınav puanlama sistemleri geliştirilirken istatistiksel analiz yöntemleri, doğal dil işleme ve yapay zekâ algoritmaları kullanılmaktadır. Doğal dil işleme ve yapay zeka yöntemlerinin otomatik puanlama sistemlerinde kullanılmaya başlanması bu sistemlerde istatistiksel analiz yapma olanağının yanı sıra biçimbilimsel, sözdizilimsel ve anlambilimsel analiz yapma olanağı sağlamıştır. Doğal dil işleme; ana işlevi bir doğal dili çözümleme, anlama, yorumlama ve üretme olan bilgisayar sistemlerinin tasarımını ve gerçekleştirilmesi konu alan bir mühendislik alanı olarak tanımlanmaktadır. Doğal dil işleme ile biçimbilimsel, sözdizimsel ve anlambilimsel analiz gibi yöntemler kullanılarak uygulamalar geliştirilmekte, araştırmalardaki durumlara ilişkin kurallar

oluşturularak problem analizi ile çözüm üretilmektedir (Nabiyev, 2012). Biçimbilimsel, sözdizilimsel ve anlambilimsel analiz doğal dil işleme yöntemleri kullanılarak yazım hatalarının denetlenmesi ve düzeltilmesi, bilgisayarlı çeviri, bilgi çıkarımı, bilgi getirimi, soru cevap sistemlerinin geliştirilmesi, özet çıkartma ve otomatik sınav puanlama gibi uygulamalar geliştirilmektedir.

Biçimbilimsel analiz, dilin yapısal özellikleri olan kelimelerin eklerini, eklerin türlerini ve kelimelerin soylarını incelemektedir. Sözdizilimsel analiz, kelimelerin diziliminin ve cümleyi oluşturan kelimelerin öğelerinin hiyerarşik uyumlarını araştırmaktadır. Anlambilimsel analiz, ayırık kelimelerin bir külliyat kullanılarak veritabanındaki uygun nesnelere eşleştirilmesini sağlamaktadır. Ayrıca anlamsal analiz ayırık kelimelerin bağlı hali getirilmesi için uygun modellerin geliştirilmesiyle ilgilenmektedir.

Anlambilimsel analiz ile otomatik puanlama sistemlerinin geliştirilmesi için doğal dil işleme metin benzerliği yöntemlerinin kullanılması gerekmektedir. Metin benzerliği, farklı iki metnin birbirine uyumunun oranını hesaplamaktadır. Otomatik sınav puanlama sistemlerinde, öğrencilerin sınav sorularına verdikleri cevap metinleri ile öğretmenlerin o sorular için hazırladığı cevap anahtarı metinlerinin uyumu metin benzerliği yöntemleriyle hesaplanmaktadır. Metin benzerliği hesaplamasının sözcüksel ve anlamsal olmak üzere iki çeşidi bulunmaktadır. Sözcüksel metin benzerliği karakter tabanlı türevleri benzerlik hesabını incelenen metinlerdeki karakter çiftlerine göre, terim tabanlı türevleri ise benzerlik hesabını metin içerisindeki kelimelerin uyumuna ve sıklığına göre metin benzerliğini hesaplamaktadır. Anlamsal metin benzerliği külliyat tabanlı türevleri benzerlik hesabını, metin kütüphanelerinde kelimelerin varoluşuna göre; bilgi tabanlı türevleri ise benzerlik hesabını anlamsal ağlara göre yapmaktadır. Metin benzerliğinde kullanılan yeni yöntemler otomatik sınav puanlama sistemlerinin başarısını arttırmaktadır.

Doğal dil işleme algoritmaları kullanılarak geliştirilen otomatik sınav puanlama sistemi ilk olarak lisede çalışan İngilizce öğretmeni Page (1966) tarafından, öğretmenlerin sınav puanlama yüklerini azaltmak amacıyla öngörülmüştür.

Page (1966) sınav puanlama işlemini, bilişim teknolojilerinin gelişmesiyle bilgisayar tarafından yapılabileceğini iddia etmiş ve bu düşüncesini sayesinde sınav puanlama projesi PEG (Page, 1968) geliştirilmiştir. PEG (Page, 1968) projesinde, eğitmen tarafından puanlanmış sınav cevaplarının sonuçları incelenerek sistem eğitilmektedir. Daha sonra sisteme girilen yeni soruların cevap puanları, önceden eğitilmiş sistemde oluşan değerlere göre hesaplanmaktadır. Araştırma sonuçlarına göre sistemin otomatik puanlaması ve eğitmenlerin puanlaması arasındaki korelasyonun 0.78 olması, sistemin yüzde yetmiş sekiz başarılı olduğunun göstergesi olarak ifade edilmiştir. Bu başarı istatistiksel yöntemlerle elde edilmiş olup, araştırmacıların otomatik puanlama sistemleri hakkında çalışma yapmalarını teşvik etmiştir.

E-rater (Burstein, 2003; Burstein ve Chodorow, 1999; Burstein ve Marcu, 2000) lisansüstü yönetime geçiş sınavının kompozisyon bölümü (GMAT-AWA) için geliştirilmiş ve kompozisyon bölümü puanlaması için 1999 Şubat tarihinde kullanılmıştır (Burstein, 2003). Nivens-Bower (2002) araştırmasında, Vantage Learning tarafından geliştirilen IntelliMetric (Elliott, 2003; Shermis ve Barrera, 2002; Shermis vd., 2003) sistemini, kişilerin etkili yazı yazma yeteneğini ölçen WritePlacer Plus sınavında kullanmıştır. Bu çalışmalar otomatik sınav puanlama sistemlerin gerçek sınavlarda kullanımını ortaya koymaktadır.

Günümüzde doğal dil işleme ve yapay zekâ algoritmalarındaki gelişmeler, otomatik sınav puanlama sistemlerinin başarısını arttırmıştır. Öğrencilerin zihinsel gelişimlerinin ölçülmesi oldukça zor, masraflı ve zaman isteyen bir iş olması, bu başarılı sistemlerin gerekliliğini göstermektedir. Bu sistemlerin ders ödevleri ve dönem içi sınavlarda kullanılabilmesi, eğitmenlerin öğrencileri değerlendirme sürecine kolaylık sağlamaktadır. Ayrıca bu sistemlerin öğrencilerin serbest zamanlı kullanımına açılmasıyla, öğrencilerin bireysel öğrenme sürecine katkı sağlamaktadır.

Bu çalışmada Türkçenin sayısal ortamlarda temsilini ve işlenmesini konu alan doğal dil işleme uygulamaları incelenmiştir. Türkçeyi konu alan doğal dil işleme araştırmaları ilgili literatür ışığında incelendiğinde bilgisayarlı çevriyazım, ses

analizleri, kavram, özet ve anlam çıkarma gibi çalışmaların gerçekleştirildiği tespit edilmiştir. Bu çalışmalar geliştirilirken Türkçenin biçimbilimsel, sözdizilimsel ve anlambilimsel analizleri kullanılmıştır. Literatür inceleme sonuçlarına göre, Türkçe otomatik sınav puanlama sisteminin olmadığı tespit edilmiştir.

Doğal dil işleme uygulamaları geliştirildikleri dile özgü olmalıdır. Bu durum dillerin biçimbilimsel ve sözdizilimsel özelliklerinin farklı olmasından kaynaklanmaktadır. Anlamsal benzerliklerin ölçülmesi için biçimbilimsel ve sözdizilimsel analizlerin de kullanılması, dile uygun işlemlerin yapılması gerekliliğini oluşturmaktadır. Dolayısıyla Türkçe otomatik sınav puanlama sistemi, Türkçe'ye özgü geliştirilmesi gerekmektedir.

Bu çalışmada uzaktan eğitim ve web tabanlı eğitim öğrenme yönetim sistemlerinde kullanılmak üzere, otomatik Türkçe sınav puanlama sistemi (TSPS) geliştirilmiştir. TSPS yazılımında, anlamsal ilişkilerinin tespiti için Türkçe WordNet ve doğal dil işleme alt yapısı için Zemberek kullanılmıştır. Ayrıca TSPS yazılımı otomatik değerlendirme modülü geliştirilmiştir.

Ölçme, Seçme ve Yerleştirme Merkezi (ÖSYM) Başkanlığı ölçme kalitesini arttırmak amacıyla 2013 yılında açık uçlu sorularla deneme sınavı gerçekleştirmiştir. ÖSYM (2013), "Açık uçlu sorularla sınavın adayların bilgi düzeylerinin daha gerçekçi ölçülmesi, şans faktörünün ortadan kaldırılması, sorunun biçiminden cevabın kestirilmesinin önlenmesi, adayın muhakeme gücünün ortaya çıkarılması gibi çok önemli faydaları yaygın olarak kabul edilirken, sınavı değerlendirmenin güçlüğü, değerlendirme süresinin uzun süreceği olması gibi olumsuz yönleri de tartışılmaktadır. Ancak, ÖSYM hem bu olumsuz yönleri ortadan kaldıracak teknolojik girişimlerde bulunarak altyapıyı oluşturmanın hem de bu altyapıyı en etkin şekilde kullanacak kurumsal yeteneği geliştirmenin gayreti içerisinde." şeklinde bir duyuru yapmıştır. TSPS yazılımının geliştirilmesi ve sonraki çalışmalarda sistemin diğer derslere de uygulanması ile ÖSYM'nin hedeflerine katkı sağlayacağı düşünülmektedir.

1.1. Tezin Amacı ve Motivasyonu

Öğrenme yönetim sistemlerinde yapay zekâ alanının alt konularından doğal dil işleme teknolojisi kullanılarak otomatik sınav puanlama sistemleri geliştirilmektedir. Bu tezin amacı, öğrenme yönetim sistemlerinde öğrencilerin öğrenme yeterliklerini ölçmek üzere kısa cevaplı açık uçlu sorular için otomatik değerlendirme yazılımı geliştirmek, bu sayede sınavların değerlendirilmesinde zaman ve ekonomik yönden kolaylık sağlamaktır.

Bu tez çalışmasında, tasarlanan TSPS yazılımı kosinüs benzerliği ve tümleşik gizli anlambilimsel analiz (TGAA) metotlarını içeren hibrid bir model kullanılmaktadır. Tez kapsamında geliştirilen TGAA metodunda eşanlı, yakınanlı, kısaltma ve rakamsal değerlerin gizli anlamsal analiz işlemesinden önce hesaplaması/düzenlemesi yapılmaktadır. TGAA metodunun işlevi, metinlerarası anlambilimsel analiz başarısının artmasını sağlamaktır.

1.2. Tez Konusunun Önemi

Geliştirilen yazılım eğitmenin oluşturduğu bilgi düzeyindeki klasik (açık uçlu) soruların cevap anahtarı metni ile öğrencilerin ilgili sorulara verdiği cevap metinlerini karşılaştırarak, metinler arası benzerliğin puanlanması işlevlerini gerçekleştirmektedir. Yazılım, hem dönem içi sınavlarda hem de dönem içi ödevlerin değerlendirilmesinde kullanılabilir. Araştırmada geliştirilen yazılımla öğrencilerin uzaktan eğitimde edindikleri e-öğrenmenin yapay zekâ yazılımıyla zenginleştirilmesi sayesinde, insan-bilgisayar etkileşimi, geri bildirim ve bireysel öğrenme konularında gelişme sağlayacaktır.

2. KAYNAK ÖZETLERİ

Bu bölümde yapay zeka, doğal dil işleme, Türkçe doğal dil işleme çalışmaları, otomatik sınav puanlama sistemi ve metin benzerliği konuları ele alınmıştır.

Yapay zekâ konusunda ilk çalışma, yapay sinir hücrelerini ve mantıksal “ve” ve “veya” işlemlerini kullanan bir hesaplama modeli öneren McCulloch ve Pitts (1943) tarafından yapılmıştır. Hebb (1949) tasarlanan sinir ağlarının öğrenme becerisini oluşturmak için, sinir hücreleri arasındaki bağlantıların şiddetlerini değiştirilmesini önermiştir. Böylece yapay sinir ağları hesaplanabilir fonksiyonlar üzerinden yeni durumlarda önceki öğrenmelerine göre işlem yapabilen, öğrenebilen bir yöntem olmuştur. Bu yöntemin alt dallarından bazıları uzman sistemler, makine zekâsı, örüntü tanıma, işaret işleme ve doğal dil işlemedir.

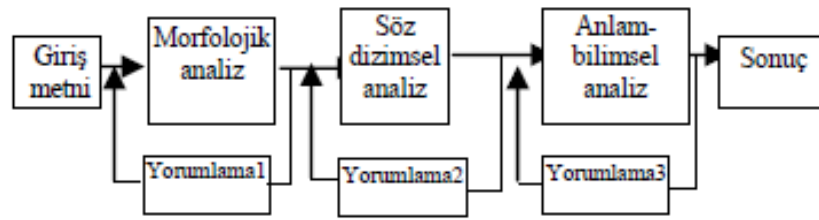
Doğal dil işleme, belli kurallar çerçevesinde oluşmuş ve gelişmiş doğal dillerin yapısının sayısal olarak çözümlenip anlaşılmasını ve yeniden üretilmesini amaçlayan yapay zekâ ve dilbilim alanlarının alt uğraş alanı olarak tanımlanır (Kışla, 2009).

Doğal dilde verilmiş metinlerin bilgisayarlı analizinde biçimbilim, sözdizilimsel, anlamsal ve problem analizi aşamaları gerçekleştirilerek;

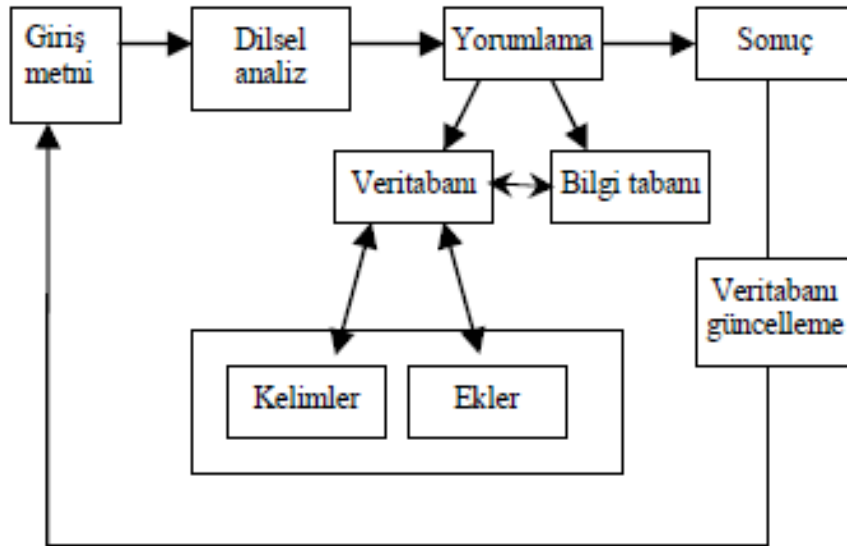
- Doğal dilde ifade edilmiş cümlelerin doğruluğu,
- Mümkün olabilecek imlâ hatalarının belirlenmesi ve düzeltilmesi,
- Girilen cümlelerin öğelere ayrıştırılmasıyla uygun sözdizilimsel yapının oluşturulması,
- Anlamsal ilişkilendirilmelerin yapılması,
- Problemi ifade eden sözdizilimsel ve anlamsal sonuçların birleştirilerek sorgulamaya uygun cevaplamanın gerçekleştirilmesi mümkün olmaktadır (Nabiyev, 2012).

Bilgisayarlar doğal dilden anlam çıkartırken kullandığı çözümlerden biri Schank'ın (1977) Turing'in Taklit Oyunu tekniğinden esinlenerek oluşturduğu

testtir. Geliştirilen test, programın sahip olduğu bilgilerle yönlendirilen soruya cevap vermesi ve cevabın sebebini belirtmesidir. Bu testte bilgisayar programı kendisine verilen bilgiyi başka sözcüklerle ifade edebilmektedir. Bilgisayar programının bu testi gerçekleştirebilmesi için işlenecek dilin matematiksel ve morfolojik analizlerinin yapılması gerekmektedir. Çakıroğlu (2006) aritmetiksel matematik problemlerinin bilgisayarla çözümünü hedeflediği araştırmasına özgü matematiksel (Şekil 2.1) ve morfolojik (biçimbilimsel) (Şekil 2.2) analizlerin aşamalarını çalışmasında sunmuştur.



Şekil 2.1. Dilin matematiksel analizi (Çakıroğlu, 2006)



Şekil 2.2. Dilin morfolojik analizi (Çakıroğlu, 2006)

Doğal dil işleme sistemlerinin çoğu metin ön işlem, biçimbilimsel analiz, söz dizimsel analiz, anlambilimsel analiz ve kullanım olmak üzere klasik dilbilim aşamalarını içerir (Stefanini ve Demazeau, 1995). Bu aşamalardan, biçimbilimsel analiz, dilin yapısal özellikleri olan kelimelerin eklerini, eklerin

türlerini ve kelimelerin soylarını incelemektedir. Kelimelerdeki eklerin araştırılması, kelimenin kendisini oluşturan kökü ve eklerinin ayrılmasıdır. Kelimenin eklerinin tespiti, eklerin oluşturacakları kelime türleri dilbilimciler tarafından önceden belirlenmiş olduğundan, kelimelerin yeni türünün belirlenmesini sağlamaktadır. Eklerin türünün araştırılması, aynı ekin farklı kelime türü oluşturması durumunda kullanılmaktadır. Eklerin farklı kelime türü oluşturması, kelimenin cümleye kattığı anlamı farklılaştırmakta olup incelenmesi gerekmektedir. Kelimelerin soylarının araştırılması, kelimelerin isim, fiil, sıfat, edat, zamir gibi hangi kelime türünden olduğunun tespit edilmesini sağlamaktadır. Ayrıca biçimbilimsel analiz ile işlenen dile özgü kurallar incelenmektedir.

Sözdizilimsel analiz, kelimelerin diziliminin ve cümleyi oluşturan kelimelerin öğelerinin hiyerarşik uyumlarının tespitini incelemektedir. Cümle içerisinde kelimelerin öğeleri özne, nesne, zarf tümleci, dolaylı tümleç, yüklem olarak ifade edilmektedir. Bu öğeler, farklı sayıda ve farklı dizilimde kullanılarak cümleler oluşmaktadır. Kelimelerin dizilimlerinin ilgili dile uygun olması, cümlelerden anlam çıkarma işlemleri için önemlidir.

Anlambilimsel analiz, ayrı kelimelerin bir külliyat kullanılarak veritabanındaki uygun nesnelere eşleştirilmesini sağlamaktadır. Ayrıca anlamsal analiz ayrı kelimelerin bağlı hali getirilmesi için uygun modellerin geliştirmesiyle ilgilenmektedir. Durum grameri ve kavramsal ilişkiler yöntemleri, kelimelerin anlamsal ilişkilendirilmesini sağlamaktadır. Durum grameri yönteminde kelimelere taşıyıcı, aygıt, yön, sebep, yer, bağış ve nesne olma durumlarına göre atama yapılmaktadır. Bu anlam atamaları cümlelerden anlam çıkarılmasını sağlamaktadır. Kavramsal ilişkiler yönteminde, kelimelerin kavramsal ilişkileri kavram ağı denilen bir modelle ifade edilmektedir. Kavram ağında kelimelerin eş ve yakın anlam durumları tutulmaktadır.

Metin ön işleme; incelenen metinlerde kırpma, küçük harfe dönüştürme, karakter olmayan simgeleri silme, etkisiz kelimeleri silme ve kök bulma işlemlerini gerçekleştirir. Ön işlemlerden;

- Kırpma işlemi metin içerisindeki kelimelerin başındaki ve sonundaki boşluğu silerek yapılmaktadır. Bu işlem cümle içerisindeki kelimelerin ayrı ayrı elde edilmesini sağlamaktadır.
- Küçük harfe dönüştürme işlemi, incelenen metin içeriğindeki tüm harflerin küçük harfe dönüştürülmesi şeklinde yapılmaktadır. Büyük küçük harf hassasiyetinden dolayı, bilgisayar “internet” ifadesi ile “INTERNET” ifadesi farklı metinler olarak algılanmaktadır. Bu yüzden tüm harflerin küçük harfe dönüştürülmesi, kelimelerin bilgisayarda doğru eşleştirilmesi için önemlidir.
- Karakter olmayan simgeleri silme işlemi, metin içerisinde var olan harf olmayan imla işaretleri gibi öğelerin silinmesiyle gerçekleştirilir.
- Etkisiz kelimeleri silme işlemi “acaba”, “ama”, “mu”, “ve”, “hem” kelimeleri gibi gereksiz kelimelerin incelenecek metin içerisinden silinmesiyle oluşturulur.
- Kök bulma işlemi metin içerisindeki kelimelerin köküne ulaşılması işlemidir. Türkçe sondan eklemeli bir dildir, Türkçede kök bulma işlemi için özel kuralların geliştirilmesi gerekir.

2.1. Türkçe Doğal Dil İşleme Çalışmaları

Bu tezde, uzaktan eğitim yazılımı olan öğrenme yönetim sistemlerinde kullanılmak üzere açık uçlu sorular için otomatik değerlendirme yazılımı doğal dil işleme yöntemi kullanılarak geliştirilmiştir. Bu bölümde TSPS yazılımının Türkçe diline uygun şekilde yazılımının hazırlanması ve çalışmanın özgünlüğünün ispatlanması amacıyla dokümantasyon metodu (Yıldırım ve Şimşek, 2000) kullanılarak literatür taraması yapılmıştır. Dokümantasyon metodunda incelenen bilimsel araştırmaların geçerli ve güvenilir olması için Türkiye’de doğal dil işleme konusunda yapılan tez çalışmaları Yüksek Öğretim Kurumunun tez tarama sayfasından incelenmiştir.

İlgili literatür taraması sonucu elde edilen çalışma örneklerinin konuları açısından değerlendirilerek;

- Biçimbilimsel analiz çalışmaları
- Sözdizimsel analiz çalışmaları
- Anlamsal analiz çalışmaları
- Problem analizi çalışmaları

olmak üzere alt başlıklarla sunulmuştur.

Elde edilen bulgular değerlendirildiğinde bu araştırma doğal dil işleme algoritmalarının eğitim teknolojilerinde kullanımına örnek oluşturacaktır. Bu tezde geliştirilen TSPS yazılımı Türkçe dilinde gerçekleştirilen ilk otomatik sınav puanlama sistemidir.

2.1.1. Biçimbilimsel analiz çalışmaları

Güngör (1995), araştırmasında Türkçe biçimbilim analizi yapmıştır. Çalışmada ek yapısı ve sıralaması incelemiş, genişletilmiş geçiş ağı formasyonunda Türkçe'nin biçimbilimsel yapısını bilgisayarda modellemiştir. Ayrıca çalışmada geliştirilen yazılım yazım kontrolü ve düzeltmesi yapabilmektedir.

Bağlılık ayrıştırması, bir tümce içindeki sözcükler arası ikili ilişkileri saptayarak o tümcenin çözümlemesini sağlayan yöntemdir. Eryiğit (2006) Türkçe'nin bağlılık ayrıştırmasını incelemiş ve modellemiştir. Çalışmada farklı nitelikte ayrıştırıcılar geliştirilerek, ayrıştırıcıların ve modellerin başarımları karşılaştırılmış ve ayırt edici öğrenmeye dayalı sınıflandırıcı tabanlı gerekirci ayrıştırıcının en iyi sonucu verdiği ortaya konmuştur.

Alkım (2006), Türkçe için doğal dil işlemede biçimbilimsel çözümleme ve sözlük tasarımı için gerçekleştirdiği sistemi bir kabuk yazılım olarak tasarlamış, sözlük ve biçimbilim kurallarını değiştirip geliştirmeye olanak sağlayan arayüzlerle

desteklemiştir. Geliştirilen bu yazılım farklı diller için de kullanılabilir özelliktedir.

Karadeniz (2007), araştırmasında Türkçe için biçimbilimsel çözümleyicinin ürettiği sonuçlardan doğru olanının saptanmasını amaçlamıştır. Çalışmada Türkçe'deki sözcüklerin belirsizlik dağılımlarını çıkarmış ve ardından sözcükler belirsizlik niteliklerine göre kümelenmiştir. Her belirsizlik türü için kurallar yazarak belirsizlikler giderilmiştir. Gerçekleştirilen çalışmanın başarımı sınanmış ve başarımının %82,6 olduğu görülmüştür.

Shylov (2008), Türkmençe ve Türkçe için iki taraflı biçimbilimsel makine çeviricisi uygulaması gerçekleştirilmiştir. Araştırmada Türkmençe ve Türkçe için biçimbilimsel çözümleyici ve biçimbilimsel üretici uygulama olarak gerçekleştirilmiştir. Türkçe ve Türkmençe için kök sözlükleri oluşturulmuştur.

Kışla (2009), araştırmasında doğal dil işleme sahasının ana problemlerinden olan biçimbilimsel çözümleme ve sözcük türü tespiti problemleri için özgün çözümler sunmuştur. Eklemeli diller için karmaşıklığı teorik olarak NPcomplete olarak bilinen biçimbilimsel çözümleme için, Türkçe'nin dilbilgisel özellikleri göz önünde tutularak basitleştirilmiş bir yöntem ortaya konmuştur. İstatistiksel ve kural tabanlı yaklaşımları bir arada kullanan yöntem çözümleme sonucunda tek bir sonuç sunması ile belirsizliğin giderilmesini de sağlamaktadır. Ayrıca önerilen yöntemin kapalı ve sınırlı bir sözlük kullanması, diğer yöntemlerden farklılaşan önemli bir özelliğidir.

Yılmaz (2009), geliştirdiği biçimbilimsel çözümleyici yapım eki almış sözcüklerin sonucunun sadeleştirilmesi, ekfiillerin biçimbilimsel analizinin tam olarak yapılması, sayısal değer içeren tarih, saat gibi sözcüklerin çözümlenmesi, kısaltma ve özel isimler için sonuç üretme ve ekler için değiştirilebilir özellikle Türk kullanıcılar için Türkçe etiketleme gibi özelliklere sahiptir. Ayrıca yazılımda çözümleme yapılmak istenen metni analiz yapılan biçime getiren metin parçalayıcı, diğer çözümleyiciler ile karşılaştırma yapabilmeyi sağlayan

karşılaştırma ve kelimeleri ek dizisinden üreten kelime türetme modülleri ile farklı özellikler gerçekleştirilmiştir.

Savaşçı (2010), çalışmasında Türkçe ve İngilizcenin bitişkenlik derecelerini istatistiksel verilerle belirlemiş olup, Türkçenin bitişkenlik derecesini 0.56 ve İngilizcenin bitişkenlik derecesini 0.12 olarak hesaplanmıştır. Dolayısıyla Türkçe’de yapılacak doğal dil işleme çalışmalarının zorluk derecesi İngilizce diline göre daha fazladır.

2.1.2. Sözdizimsel analiz çalışmaları

Hallaç (2007), verilen bir Türkçe dokümandaki her sözcüğe, en uygun sözcük sınıfını atamayı amaçlayan Türkçe için geliştirilen kural tabanlı bir Sözcük Türleri Belirleyicisi olan TurPOS’u geliştirmiştir. TurPOS, girdi dokümanı olarak biçimbilimsel çözümlene uygulaması tarafından üretilen bir derlem kullanmaktadır. Sistem ayrıca, Türkçe gramer kurallarını içeren bir kural dosyası kullanmaktadır.

Agun (2008), araştırmasında Türkçe’deki sözdizimsel özelliklerin öğrenilmesi için çizge tabanlı bir otomatik öğrenme modeli sunmuştur. Çalışmada bir derlem kullanılarak tasarlanan çizge modeli eğitilmiş ve girilen bir cümle için doğru sözdizimsel etiketler bu model aracılığıyla çıkarılmıştır. Modelin tasarımı sırasında, olasılık tabanlı çizge modeli olan Saklı Markov Modelleri ve çizge teorisinden yararlanılmıştır. Sunulan çalışmada diğer olasılık tabanlı etiketleme algoritmalarından ve istatistiksel doğal dil işleme çalışmalarından farklı olarak Türkçe'nin biçimbilimsel özelliklerinin de kullanılabildiği olasılık tabanlı bir çizge modeli geliştirilmiştir.

Eker (2009), doğal dil işlemenin temel problemlerinden sözdizimsel ayrıştırma ile ilgili çalışmıştır. En yaygın iki ayrıştırma, öbek yapısı ayrıştırma ve bağımsallık ayrıştırma değildir. Çalışmada metinsel gerektirimleri ile ayrıştırıcıların değerlendirilmesi yapılmış, öbek yapısı ayrıştırıcı en yüksek puanı almıştır.

Şentürk (2009), çalışmasında Türkçe dili için kural tabanlı yani biçimbilimsel bir bul ve değiştir işlevi yapılmasına çalışmıştır. Bu çalışmaya gerek duyulmasının en önemli nedeni yazım programlarının Türkçe dili için yetersiz olmasından kaynaklanmaktadır. Araştırmada sonlu durum makineleri oluşturularak tüm kök ve ek türleri için doğru çözümlenmeler yapılması sağlanmıştır. Bulunan sözcüklerin eklerinin doğru çözümlenmesi, bu çözümlenen ekler ile de doğru değiştirme yapılması sağlanmıştır.

Kutlu (2010), Türkçe metinler için bir isim tamlaması çıkarımı sistemi geliştirmiştir. Araştırmada cümle bileşenleri arasındaki ilişkiyi göstermek ve isim tamlamalarını bulmak için ağırlıklı bir kısıtlayıcı bağımlı çözümleyici kullanılmıştır.

Tahiroğlu (2010), araştırmasında bilgisayar destekli sözlük bilimi çalışmalarında kullanılmak üzere web tabanlı derleme sözlüğü oluşturmuştur. Oluşturulan derleme sözlüğünün veri tabanını MySQL ve web ara yüzünü PHP ile kodlamıştır. Geliştirilen yazılımın gelecek sözlük bilimi araştırmalarına ve sözlük yazımlarına, çevrim içi, paylaşımlı sözlüklerin hazırlanması aşamalarında veri sağlayabileceği düşünülmektedir.

Turna (2011), çalışmasında Türkçedeki sözcük öbeklerini tanımlamak için kural tabanlı bir sistem geliştirilmiş ve sistemi kullanan bir yazılım hazırlanmıştır. Geliştirilen yazılımla sözcük türlerinin belirlenmesi için Türkçe kelime yapılarının tanımlanması sağlanmaktadır.

Zafer (2011), Türkî diller için genel bir cümle çözümleyici geliştirmiştir. Geliştirilen çözümleyici bağlamdan-bağımsız gramer kurallarına, morfolojik çözümlenmeye ve geçerlilik kurallarına dayanmaktadır. Çalışmada Türkçe ve Türkmençe'nin grameri bilgisayarlı açıdan incelenmiş ve sistem Türkçe için hayata geçirilmiştir. Daha sonra kısmi değişiklikler ile Türkmençe'ye uyarlanmıştır.

Kazkılınç (2012), haber metinlerinde, haber metninin öznesi, yüklemi, yer ve zamanını belirtecek söz öbeklerinin metinde bulunup, metnin etiketlenmesini gerçekleştirmiştir. Bu amaçla araştırmada, metinde geçen cümleler içerisinde seçilen en baskın özne, yüklem, yer ve zaman bilgileri çıkarılmıştır. Çalışmada elde edilen bu etiket bilgileri sayesinde metnin konusu temsil edilmekte olup, anlamsal ağda etiket olarak ve arama motorlarında istenilen veriye ulaşabilmek için kullanılabilir.

2.1.3. Anlamsal analiz çalışmaları

Orhan (2006), Türkçe metinlerde anlam belirsizliği olan sözcüklerin anlamlarının belirginleştirilmesini sağlayacak en uygun algoritmaların ve özelliklerin belirlenmesini ele almıştır. Türkçe için sözcükte anlam belirginleştirilmesi çalışmalarında kullanılacak sözcükler ve anlam sınıfları oluşturulmuş, algoritmalarda kullanılacak metinlerin elle işaretlenmesi gerçekleştirilmiş ve kavramsal bir sözlük hazırlanarak bu alanda yapılacak araştırmalara önemli bir katkıda bulunulmuştur.

Bahadır (2007), araştırmasında ilköğretim öğrencileri seviyesindeki aritmetik matematik problemlerinin çözümlenmesini ve simgelenmesini gerçekleştirmiştir. Geliştirilen uygulamanın modülleri; kelimelerin biçimbilimsel analizinin yapılması, kelimelerin aldıkları eklerin belirlenmesi, soru kalıbının belirlenmesi, sorunun çözülmesi, çözümün denklemler şeklinde ifade edilmesi, sorunun ve çözümün simgelenmesi olarak sunulmuştur.

Birant (2008), Aktaş (2006) tarafından geliştirilen ve doğruluk oranı %99'u bulan cümle sonu bulma algoritmasını temel alarak ve bu çalışmanın üzerinde yapılan değişikliklerle kullanılan bazı yapıları standart bir hale getirmiştir. Araştırmada bu standardizasyon çalışmaları hedeflenen biçimbilimsel çalışma programının sonucu olarak kullanıcıya sunulan ve biçimbilimsel ayrıştırma sırasında ekler arası ardalanma kurallarını içeren dosyalarda XML yapısı kullanılmıştır. XML yapılarında yer alacak ek çeşitlerinin ise isimleri, belirlenen

bir takım kurallar çerçevesinde sistematik olarak kısaltılarak kök ve eklerin etiketlenmesine imkân sağlanmıştır.

Özdemir (2009), Türkçe metinler içerisinde anlam belirsizliğine sahip olan sözcükler için uygun özellik ve algoritmaların seçilerek, anlam belirginleştirme işlemi üzerine çalışmıştır. Araştırmada çalışmaların yapılacağı işaretlenmiş metinlerin eksikliği sebebiyle öncelikle seçilen örnek kelimeleri içeren cümlelerden oluşan veriler toplanmıştır. Daha sonra seçilen kelimeler için anlamın ayırt edilmesini sağlayan özellikler belirlenmiş ve Öğreticili Öğrenme algoritmaları verilere uygulanmış, değerlendirme yöntemleri ile elde edilen sonuçlar değerlendirilmiştir. Çalışmada yapılan test işlemlerinde anlam belirginleştirme için NaiveBayes, Kstar, SimpleCart ve Bagging algoritmaları kullanılmıştır. Ayrıca kelimeler için çalışmada etkili olabilecek özellikler belirlenmiştir. Seçilen bu algoritmaların ve özelliklerin ne kadar etkili oldukları ortaya konmuştur.

Adalı (2009), Türkçe belgelerin otomatik olarak işlenmesi ve bu belgelerden bilgi çıkarımı üzerine çalışmıştır. Geleneksel bilgi çıkarımı sistemleri, giriş metnini sıralı kelimeler olarak ele alıp genellikle anlamsal özellikleri üzerine yoğunlaşırken, önerilen mimari yardımı ile belge yapısının sağladığı ipuçlarından yararlanılmaktadır. Bununla birlikte, belge tutarlılığını sağlamak için, belgede yer alan varlıklar arasındaki ilişkiler sınanmakta ve çıkarımı yapılmış varlıklar gerçekte kullanılan veriler ile karşılaştırılmaktadır. Önerilen tümleşik mimari; Türkçe için biçimbilimsel analiz modülü, belge yapı analiz modülü, alan ontolojisi, çıkarım ontolojisi içermektedir.

Kalender (2010), araştırmasında belgeler için otomatik olarak anlamsal etiketler oluşturan bir sistem önermektedir. Çalışmada UNIPedia çağdaş (web de geçen güncel kelimeler) referansları içeren bir bilgi tabanı sağlamaktır. UNIPedia çeşitli ontolojik bilgi tabanlarını WordNet kavramlarıyla ilişkilendirmektedir. Güncel ve güvenilir bilgi içeren Wikipedia ve OpenCyc bilgi tabanları WordNet kavramları ile eşleştirilmiştir. Bilgi tabanlarını

ilişkilendirmek için kavramların ontolojik ve istatistiksel özelliklerini kullanan kural tabanlı sezgiseller kullanılmıştır.

Per (2011), araştırmasında Türkçe için kavram çıkarma sistemi ortaya koymuştur. Türkçe karakterlerin bilgisayar diline uymaması ve Türkçe'nin sondan eklemeli karmaşık yapısından dolayı öncelikle bir ön işleme aşaması gereklidir. Ön işlemenin sonucunda, çekim eklerinden de ayrılmış olan kelimelerin sadece isim türünde olanları kullanılmıştır. Sistem üretmesi gerekenden daha fazla kavram üretmiş olmasına rağmen, yüzde 51 başarı ile dokümanlara ait kavramları bulmuştur. Kavramların yapı itibarıyla dokümanlarda aynen geçmeme ihtimali ve Türkçe'nin karmaşık yapısı düşünülürse bu sonuç oldukça başarılı olarak değerlendirilebilir.

2.1.4. Problem analizi çalışmaları

Akman (2004), pahalı ve yavaş bir yöntem olan elle çevriyazım ile hızlı ve hata oranı yüksek olan bilgisayarlı çevriyazım yöntemlerini çalışmasında birleştirmiştir. Çevriyazılacak konuşmalarla ilgili hipotezleri içeren ve bir konuşma tanıma motorunun çıktısı olan örüler, harf tabanlı, gerekirci, ağırlık sonlu durum alıcılarını dönüştürülmüş, içerik bakımından konuşma verisiyle örtüşen bir metin derlemiyle eğitilmiş ve harf tabanlı istatistiksel bir dil modeliyle (Mackay, 2013) birleştirilmiştir.

Eş (2005), elektronik postaları otomatik olarak yasal veya istenmeyen kategorilerine ayırabilen bir filtreleme konusu çalışmıştır. Araştırmada elektronik postaları otomatik olarak yasal veya istenmeyen şeklinde sınıflandırmak için bazı makine öğrenim metotları ve bilgi elde etmenin bazı fikirlerinden faydalanarak, elektronik posta filtreleme işinde etkin sonuçlar elde edilmeye çalışılmıştır.

Yıldırım (2005), araştırmasında sms mesajlarının istatistiksel doğal dil işleme yöntemleri kullanılarak anlamlandırılmıştır. Çalışmada sms mesajları ile kullanıcıya sunulan servislerin önceden belirlenmiş anahtar kelimelerle

çalıştığı, kullanıcıların bu anahtar kelimeleri yanlış girmeleri takdirde bu servisi alamadıklarından yola çıkılarak, istatistiksel doğal dil işleme yöntemlerinin başında gelen N-Gram yöntemiyle bu probleme bir çözüm yöntemi sunulmuştur.

Tülek (2007), Türkçe için metin özetleme çalışması gerçekleştirmiştir. Yazılımda Türkçenin yapısı göz önüne alınarak, bir metnin özetlenmesi için farklı istatistiksel yöntemler tanıtılıp yazılımla gerçekleştirilmiş ve bu yöntemlerin Türkçeye uygunluğu tartışılmıştır. Diğer tüm Türkçe bilgi erişimi sistemlerinde de gerekli olduğu gibi, Türkçenin sondan eklemeli yapısının gözetilmesi amacıyla farklı gövdeleme algoritmalarının özetleme başarımına etkisi incelenmiştir. Başarımlarının daha yüksek olması amacıyla, gerçekleştirilen gövdeleme algoritmalarında sözcüklerin olası kök ve ek birleşimlerini üreten biçimbilimsel çözümleyici kullanılmıştır. Gövdelenmiş bu sözcükler farklı özetleme yöntemleri aracılığıyla incelenip her yöntem için özetle yer alacak cümleler belirlenmiştir. Daha sonra bu yöntemlerin ürettiği sonuçlar birleştirilerek son özet oluşturulmuştur.

Delibaş (2008), doğal dil işleme ile Türkçe yazım hatalarının denetlenmesini konu alan çalışmasında, girilen metindeki cümlelerdeki kelimelerin köklerinin bulunması, eklerinin ayrıştırılması, kelimenin doğruluğunun sınanması, yanlış yazılmış kelimelere sözcük önerilmesi ve Türkçe olmayan yabancı kelimelerin sözlüğe eklenmesi işlemleri gerçekleştirilmiştir.

Yılmaz (2008), araştırmasında lise öğrencilerini ve üniversite adaylarını hedef olarak Türk Üniversiteleri hakkında bilgileri işleyen anlamsal ağ için doğal dil ara yüzü prototipi geliştirilmiştir. Geliştirilen yazılıma üniversiteler hakkında cümleler halinde soru sorulduğunda cevaplar ekrana yansımakta olup, öğrencilerin üniversite tercih dönemine yardımcı bir program hazırlanmıştır.

Aksoy (2008), araştırmasında doğal dil işleme ve genetik algoritma kullanarak üç boyutlu sahne oluşturmuştur. Çalışmada önceden tanımlanan doğal dille kodlanmış anlam çerçevesi içerisinde verilen nesnelere ve nesnelere arası

ilişkileri girdi olarak alan, çalışmada savunulan yerleşim kurallarını uygunluk değerlendirmesinde kullanan genetik algoritma uygulamasıyla 3 boyutlu sahne yorumunu üreten bir uygulama yazılımı sunulmuştur.

Köprü (2008), araştırmasında ses çevirisi amacıyla otomatik ses tanıma ve otomatik çeviri sistemlerini entegre eden özgün bir yaklaşım sunmuştur. Sunulan yöntem, kelime grafiği formatındaki ses verilerini işleyebilen ilk kural tabanlı otomatik çeviri sistemini içermesi açısından özgündür. Çalışmada araştırmada geliştirilen bağdaştırma yönteminin ilk-en iyi ve N-en iyi tekniklerinden daha iyi performans gösterdikleri de ortaya koyulmuş, argümanların doğruluğu deneylerle kanıtlanmıştır.

Aranan bilgiyi içeren belgelerin bulunması için web arama motorları kullanılmaktadır. Ancak, birçok durumda kullanıcı bir belge kümesinden çok belirli bir bilgiye ihtiyaç duyar. Soru yanıtlama sistemleri bu problemi adreslemektedir. Soru yanıtlama sistemleri bir sorunun yanıtı olarak bir belge kümesi yerine açık yanıtlar döndürürler. Er (2009), çalışmasında Türkçe tekil yanıtli soru yanıtlama için örüntü eşleştirme yaklaşımı ile bir yazılım geliştirmiştir.

Görmez (2009), makine öğrenme algoritmalarıyla Türkçe metin seslendirme sistemi yazılımı gerçekleştirmiştir. Sistem direk kaydedilen ve sürekli bir konuşmadan kesilen birimlerden oluşan iki farklı ses veritabanına sahip olup, yazılım birimlerin konuşmadan elle ve otomatik olarak kesilmesine imkan vermektedir. Ses sinyalinin sıfırdan geçiş sayısı ve sesin enerjisi gibi sinyal özellikleri otomatik kesme işlemi için kullanılmıştır.

Soysal (2010), doğal dil işleme ve bir alan ontolojisi yöntemlerini kullanarak Türkçe radyoloji raporlarından yapılandırılmış bilgi elde etmiştir. Alan ontolojisi bilgi çıkarımı sırasında, varlık tanımlama ve ilişki çıkarımı aşamalarında kullanılmıştır. Çalışmanın doğal dil işleme işlemlerinde, Türkçe morfolojik olarak zengin bir dil olduğu için, sistem bir morfolojik analizör olarak kullanılmış ve çıkarım kurallarında bu morfolojik özelliklerden faydalanılmıştır.

Radyoloji bölümleri hastaların vücutlarını görselleştiren teknikler kullanır ve bu görüntüler doktorlar tarafından incelenerek düz metin raporlara dökülür. Medikal bilgi sistemleri açısından bu düz metin raporlardaki bilginin çıkartılması önem arz eder. Hadımlı (2011), biri kural tabanlı biri veri tabanlı olmak üzere Türkçe radyoloji raporlarının işlenmesinde kullanılabilir iki yöntem önermektedir. Önceki medikal doğal dil işleme çalışmalarının aksine, her iki yöntemde de bir medikal sözlük ya da bir medikal ontoloji kullanılmamaktadır. Bilgi çıkarımı, verilen cümle içinde medikal olarak ilişkili sözcük öbeklerinin ve bunların ilişkilerinin belirlenmesi seviyesinde yapılmaktadır. Amaç, diğer etkenlerin yokluğunda, Türkçe'nin özelliklerinin medikal bilgi çıkarımı ve erişimi için sunabileceği referans performansı belirlemektir.

Albayrak (2011), Türkçe'nin kullanımı ile psikolojik durum arasındaki ilişkiyi araştırmıştır. Araştırmada depresyonlu, depresyonsuz, anksiyeteli, anksiyetesiz kişilerden Türkçe yazılar toplanmıştır. Yazılar her bir kelime, her bir tanı grubu tarafından en çok kullanılan kelime grupları, kipler, kişi zamirleri, fiiller ve isimler gibi özelliklerin kullanımına göre analiz edilmiştir. Tanısı konulmuş kişiler arası kelime kullanımı farklarını test etmek için bir program geliştirilmiştir. Test programının sonuçları, Türkçe'de kelime kullanımının psikolojik durum hakkında birçok ipucu verdiğini göstermiştir.

2.2. Otomatik Sınav Puanlama Sistemi

Otomatik sınav puanlama sistemi ilk olarak lisede çalışan İngilizce öğretmeni Page (1966) tarafından öğretmenlerin sınav puanlama yüklerini azaltmak amacıyla öngörülmüştür. Otomatik sınav değerlendirme sistemlerinden en çok bilinenler sınav puanlama projesi PEG (Page, 1966; 1968; 2003), sınav değerlendirici IEA (Landauer vd., 2003), e-rater (Attali ve Burstein, 2006; Burstein, 2003), IntelliMetric™ (Rudner vd., 2006) ve BETSY (Rudner ve Liang, 2002)'dir.

PEG (Page, 1968) sistemi eğitim aşaması ve puanlama aşaması olmak üzere iki aşamadan oluşmaktadır. PEG ilk aşamada örnek sınavları kullanarak sistemi eğitir ve sonraki aşamada eğitim sonucu elde edilen değerlerle yeni sınavların puanları öngörür. Sonuçta, her sınav için eğitim aşamasından hesaplanan beta ağırlıklarına göre yeni sınavlara puan istatistiksel hesaplama ile atanır (Chung ve O'Neil, 1997). PEG sisteminin eğitimi için yüz ile dört yüz örnek sınav gerektirmektedir. PEG sisteminin başarısının test edildiğinde, sistemin puanlaması ile öğretmenlerin puanlaması arasındaki ilişki R korelasyonuna göre 0.78 olarak bulunmuştur. Aynı çalışmada iki veya daha fazla öğretmenin puanlamaları arası R korelasyonu ise 0.85'dir (Kukich, 2000).

IEA (Landauer vd., 2003) sistemi gizli anlambilimsel analiz (Lemaire ve Dessus, 2001) semantik metin analizi metodunu kullanmaktadır. IEA sistemi bir sınavı analiz etmek için üç kaynak kullanmaktadır; daha önceden puanlanmış sınavlar, uzman model sınavlar ve bilgi kaynak materyaller, puanlanmamış sınavların iç karşılaştırılması (Landauer vd., 2003). Bu yöntem her sınavı benzer metinlerle içerik kalitesi açısından karşılaştırılmasını sağlamaktadır (Landauer vd., 2000; Landauer vd., 2003; Streeter vd., 2004). Öncelikle IEA içerik benzerliğini bir öğrencinin puanı ile öğretmenler tarafından puanlanan diğer sınavlarla karşılaştırarak puanlamalarının ne kadar uyumlu olduğunu belirlemektedir (Landauer vd., 2000; Rudner ve Gagne, 2001; Streeter vd., 2004). Sistem sonraki aşamada tüm puanları tahmin etmektedir (Landauer vd., 2000).

E-rater lisansüstü yönetime geçiş sınavı kompozisyon bölümü (GMAT-AWA) (Burstein, 2003; Burstein ve Chodorow, 1999; Burstein ve Marcu, 2000) için geliştirilmiş ve kompozisyon bölümü puanlaması için 1999 Şubat tarihinde kullanılmıştır (Burstein, 2003). E-rater değerlendirme modeli geliştirmek için külliyat tabanlı doğal dil işleme aracı kullanmaktadır. Öğretmenler tarafından değerlendirilmiş örnek eğitim sınavlar kullanılarak kelime frekans vektörü, kelime frekansı ve kelime ağırlığı dikkate alınarak ağırlık vektörü oluşturulur. Elde edilen ağırlık vektörüne göre sisteme gelen yeni sınavlar yazılım tarafından puanlandırılır. E-rater sistemin eğitilmesi için önceden puanlandırılmış en az dört yüz altmış beş sınav gerekmektedir.

Burstein vd. (1998) e-rater sisteminin performansının incelemek amacıyla yaptıkları çalışmada, örneklem olarak beş yüz GMAT ve iki yüz İngilizce yazma becerisini ölçen TWE'de kullandıkları çalışmada iki eğitmenin puanlamalarını incelemişler ve puanlama korelasyonu 0.82-0.89 aralığında bulunmuştur. Aynı çalışmada, eğitmen1 puanı ile e-rater sistemi arasındaki puanlama 0.80-0.87, eğitmen2 puanı ile e-rater arasındaki puanlama 0.79-0.87 korelasyonunda bulunmuştur (Burstein vd., 1998).

Criterion, e-rater ve Critique teknolojilerini kullanan web tabanlı sınav puanlama ve değerlendirme sistemidir. Critique, yazım hata denetimi yapan ve sınavda istenen/istenmeyen unsurları tanıyan bir sistemdir. Critique, özellikleriyle eğitmenlere öğrencilerin gerçekleştirdiği web ortamındaki öğrenme aktivitelerini (yazım denetimi, öğrencilerin eğitim sistemine giriş-çıkış zamanlamaları vb. tanımlayıcı bilgiler) elektronik portfolyo gibi izleme imkânı sunmaktadır (Burstein vd., 1998).

IntelliMetric, Vantage Learning tarafından geliştirilen otomatik sınav puanlama sistemidir ve ilk kez yapay zeka kullanılan sınav puanlandırma aracı içermektedir (Elliott, 2003; Shermis ve Barrera, 2002; Shermis vd., 2003). E-rater gibi IntelliMetric yazılımı da doğal dil işleme ile geliştirilmiştir. IntelliMetric sisteminin eğitilmesi için en az 300 adet sınav kâğıdının alanında uzman kişiler tarafından değerlendirilmesi gerekmektedir. Kişiler tarafından puanlanan sınavların metin özelliklerini dikkate alınarak gizli anlambilimsel boyutlar yöntemiyle (Vantage Learning, 2003a) kategoriler oluşturmaktadır. Sistemde odak ve anlam bütünlüğü, organizasyon, gelişme ve detaylandırma, cümle yapısı, teknik ve üslup olmak üzere beş kategori elde edilmekte ve bu beş kategoriden alınan puanlara göre genel puan verilmektedir (Vantage Learning, 2003b, 2003c). MyAccess, Vantage Learning'in IntelliMetric otomatik sınav puanlama sistemi ile çalışan web-tabanlı yazı değerlendirme aracıdır.

Nivens-Bower (2002) araştırmasında IntelliMetric sisteminin performansını WritePlacer Plus sınavını 30 öğrencide uygulayarak incelemiştir. Sınav iki

eğitmen ve IntelliMetric tarafından değerlendirilmiştir. Araştırmada grupların ortalamaları karşılaştırmak amacıyla bağımlı-örneklem t-testi ve puan frekansı dağılımını incelemek amacıyla Wilcoxon işaret-sıralama testi kullanılmıştır. Araştırmada bağımlı-bağımlı örneklem t-testi sonucunda anlamlı bir farklılık bulunmadığı ve Wilcoxon işaret-sıralama testi sonucunda yine anlamlı bir farklılığın bulunmadığı belirtilmiştir. Çalışma sonuçlarına göre IntelliMetric sistemi puanlamasının eğitmenlerin puanlaması ile uyumlu olduğu belirtilmiştir.

BETSY, Rudner ve Liang (2002) tarafından geliştirilen Bayes teoremini temel alan otomatik puanlama sistemidir. BETSY yeni dokümanları sınıflamak için daha önceden eğitilmiş 1000 adet metin gerektirir (Rudner vd., 2005). Sınıflama işlemi aşamaları; eğitilmiş kelimeler alınması, veri tabanı istatistiklerini değerlendirilmesi, genel olmayan kelimelerin elenmesi, gereksiz kelimelerin tespiti, kelime çiftlerinin tespiti, veri tabanı istatistiklerinin tekrar değerlendirmesi, genel olmayan kelimelerin elenmesi ve eğitilen kelime grubunun puanlandırılması ve sınıflandırılmayan eğitilmiş kelimelerin kırılması olarak tanımlanmaktadır (BETSY, 2014). Çizelge 2.1’de incelenen otomatik puanlama ve değerlendirme sistemleri geliştirici, teknik, çözümleme, eğitim-öğretim uygulaması durumu ve sistemin eğitilmesi için gerekli olan sınav sayısı özellikleri yer almaktadır.

Çizelge 2.1. Otomatik sınav puanlama sistemlerinin karşılaştırılması

Otomatik Sınav Puanlama Sistemi	Geliştirici	Teknik	Çözümleme	Eğitim-Öğretim Uygulaması	Sistemin Eğitilmesi İçin Gerekli Olan Sınav Sayısı
PEG	Page (1966)	İstatistik	Biçimbilimsel	Yok	100-400
IEA	Landauer vd. (2000)	GAA	İçerik	Yok	100-300
E-rater	Burstein vd., (1998)	Doğal Dil İşleme	Biçimbilimsel ve İçerik	Criterion	465
IntelliMetric	Vantage Leaning (2014)	Doğal Dil İşleme	Biçimbilimsel ve İçerik	My Access	300
BETSY	Rudner ve Liang (2002)	Bayes Sınıflama	Biçimbilimsel ve İçerik	Yok	1000

Leacock ve Chodorow (2003) kısa cevaplı soruların cevaplarını puanlandırmak amacıyla C-rater isimli otomatik puanlandırma motoru geliştirmişlerdir. Bu yazılımda sadece kelimelerin birebir kontrolü yerine, kelimelerin eş anlamlılık ve morfolojik analizi gibi ek işlemlerden geçirilmesiyle programın geçerliliğini arttırmışlardır. Yazılımın uygulandığı çalışmalarda eğitimcilerin puanlaması ve C-rater yazılımının puanlaması karşılaştırıldığında yüzde seksen dördlük bir benzerlik saptanmıştır.

Pulman ve Sukkarieh (2005) çalışmalarında Oxford-UCLES isimli otomatik kısa cevap puanlandırma sisteminde endüktif mantık programlama, karar ağaçları ve bayes öğrenmesi gibi makine öğrenme tekniklerini kullanmışlardır. Yazılımda anahtar kelimeler ve eş anlamlı kelimeler öznitelik olarak belirlenmiştir.

Pérez vd. (2005) geliştirdikleri yazılımda gizli anlambilimsel analiz ve Papineni vd. (2001) önerdiği n-gram puanlandırma olan BLUE isimli algoritma kullanarak otomatik değerlendirme sistemi geliştirmişlerdir. Araştırma bu iki algoritmanın beraber çalıştırılmasıyla daha geçerli bir değerlendirme yazılımı geliştirdiklerini vurgulamışlardır.

Mohler ve Mihalcea (2009) araştırmalarında bilgi ve anlam tabanlı metin benzerliği yöntemlerini karşılaştırarak danışmansız tekniklerle otomatik kısa cevap puanlama sistemi gerçekleştirmişlerdir. Ayrıca çalışmada öğrencilerin sorularına otomatik geri dönüt veren bir yazılım geliştirmişlerdir.

İlkuçar (2015) araştırmasında, metin tabanlı programlama dersi ödevlerinin benzerlik oranlarının belirlenip, buna göre otomatik değerlendirme yapılabildiği bir yazılım n-gram yöntemi ile gerçekleştirmiştir.

Aşağıda otomatik değerlendirme yapan yazılımlardan en çok öne çıkanlar listelenmiştir.

- Project Essay Grade (Page 1966, Page 1994)
- Educational Testing Service 1 (Burstein vd., 1996)
- Text Categorisation Technique (Larkey, 1998)
- Intelligent Essay Assessor (Landauer, 1998)
- Electronic Essay Rater (Burstein vd., 1998)
- Schema Extract Analyse and Report (Christie, 1999)
- Intelligent Essay Marking System (Ming vd, 2000)
- Conceptual Rater (Burstein vd., 2001)
- Bayesian Essay Test Scoring System (Rudner vd. , 2002)
- AutoMark (Mitchell vd., 2002)
- Paperless School Free Text Marking Engine (Mason ve Grove-Stephenson, 2002)
- Blue Wren Software (Williams, 2012)
- SAGrader (Idea Works, 2013)

Otomatik değerlendirme sistemi konusunda literatürde yapılan çalışmalar incelendiğinde; anlam tabanlı metin benzerliği yöntemlerinin bilgi tabanlı metin benzerliği yöntemlerine göre daha başarılı olduğu, makine öğrenme algoritmalarının otomatik puanlama sistemlerinde kullanımı, bazı metin benzerliği algoritmalarının tümleşik kullanımının otomatik puanlama sistemi oluştururken kullanıldığı, yapılan araştırmalarda eğitimci puanlaması ile yazılım puanlamasının arasındaki korelasyon hesabı ile yazılım performansının hesaplandığı tespit edilmiştir.

Geliştirilen otomatik değerlendirme yazılımları, ilgili çalışmanın yapıldığı ülkenin diline özgü olarak hazırlanmıştır. Türkçe dilinde bu projenin gerçekleştirilmesiyle, Ülkemizdeki uzaktan eğitim derslerinde ve serbest zaman çalışması olarak öğrencilerin kullanımına açılmasıyla eğitim-öğretime katkı sağlayacağı düşünülmektedir.

2.3. Metin Benzerliđi

Herhangi iki belge arası benzerliđin sözcüksel veya anlamsal olarak hesaplanması metin benzerliđi yöntemleriyle gerçekleştirilir. Sözcüksel metin benzerliđi karakter ve terim tabanlı olmak üzere farklı yöntemlerle hesaplanmaktadır. Anlamsal metin benzerliđinin ise külliyat ve bilgi tabanlı olmak üzere iki farklı türü vardır.

2.3.1. Sözcüksel metin benzerliđi

Sözcüksel metin benzerliđi türünden karakter tabanlı türevleri benzerlik hesabı, incelenen metinlerdeki karakter çiftlerine göre yapmaktadır. Karakter tabanlı ile metin benzerliđi en uzun ortak alt dizi (LCS) algoritmasının yanı sıra, Damerau-Levenshtein (Hall ve Dowling, 1980; Peterson, 1980) Jaro (Jaro, 1989; Jaro, 1995), Jaro-Winkler (Winkler, 1990), Needleman-Wunsch (Needleman ve Wunsch, 1970), Smith-Waterman (Smith ve Waterman, 1981) ve N-gram (Alberto vd., 2010) yöntemleriyle hesaplanmaktadır.

Terim tabanlı benzerlik ise metin içerisindeki kelimelerin uyumuna ve sıklığına göre metin benzerliđini hesaplamaktadır. Terim tabanlı benzerlik Manhattan uzaklığı (Eugene, 1987), Kosinüs benzerliđi (Mihalcea vd., 2006), Dice'nin benzerlik katsayısı (Dice, 1945), Öklit uzaklığı, Jaccard benzerliđi (Jaccard, 1901), eşleme benzerlik katsayısı ve örtüşme benzerlik katsayısı olmak üzere farklı yöntemlerle hesaplanabilmektedir.

2.3.2. Anlamsal metin benzerliđi

Anlamsal metin benzerliđi türünden külliyat tabanlı türevleri benzerlik hesabı, metin kütüphanelerinde kelimelerin varoluşuna göre yapmaktadır. Külliyat tabanlı metin benzerliđi, hiperuzay analog dili (hyperspace analogue to language-HAL) (Lund vd., 1995; Lund ve Burgess, 1996), gizli anlambilimsel analiz (latent semantic analysis-LSA) (Landauer ve Dumais, 1997), genelleştirilmiş gizli anlambilimsel analiz (generalized latent semantic analysis-

GLSA) (Matveeva vd., 2005), açık anlambilimsel analiz (explicit semantic analysis-ESA) (Gabrilovich ve Markovitch, 2007), diller arası açık anlambilimsel analiz (cross-language explicit semantic analysis-CLESA) (Martin vd., 2008), noktasal karşılıklı bilgi - bilgi alma (pointwise mutual information-information retrieval-PMI-IR) (Turney, 2001), ikinci dereceden eş-oluşum noktasal karşılıklı bilgi (second-order co-occurrence pointwise mutual information) (Islam ve Inkpen, 2006; Islam ve Inkpen, 2008), normalleştirilmiş google mesafe (normalized google distance-NGD)(Cilibrasi ve Vitanyi, 2007), birliktelik kullanarak benzer kelimeleri dağıtımsal ayıklama (extracting distributionally similar words using cooccurrences-DISCO) (Peter, 2009) yöntemleriyle hesaplanmaktadır.

Bilgi tabanlı benzerlik ise anlamsal ağlara (Mihalcea vd., 2006) göre metin benzerliğini hesaplamaktadır. Anlamsal ağların en popüler olan, WordNet (Miller vd., 1990) kelimeler hakkında eş anlam, zıt anlam, tanım gibi özellikleri barındırması özelliğiyle, bilgi tabanlı benzerlik için kullanılmaktadır. Sözcüksel ve anlamsal metin benzerliklerinin farklı kombinasyonlarıyla hibrid benzerlik ölçüleri oluşturulabilmektedir.

3. YÖNTEM

Bu bölümde otomatik Türkçe sınav puanlama sistemi (TSPS) yazılımının geliştirilmesi, geliştirilen TSPS yazılımının hesaplama modülü ve TSPS yazılımının değerlendirilmesi konuları ele alınmıştır.

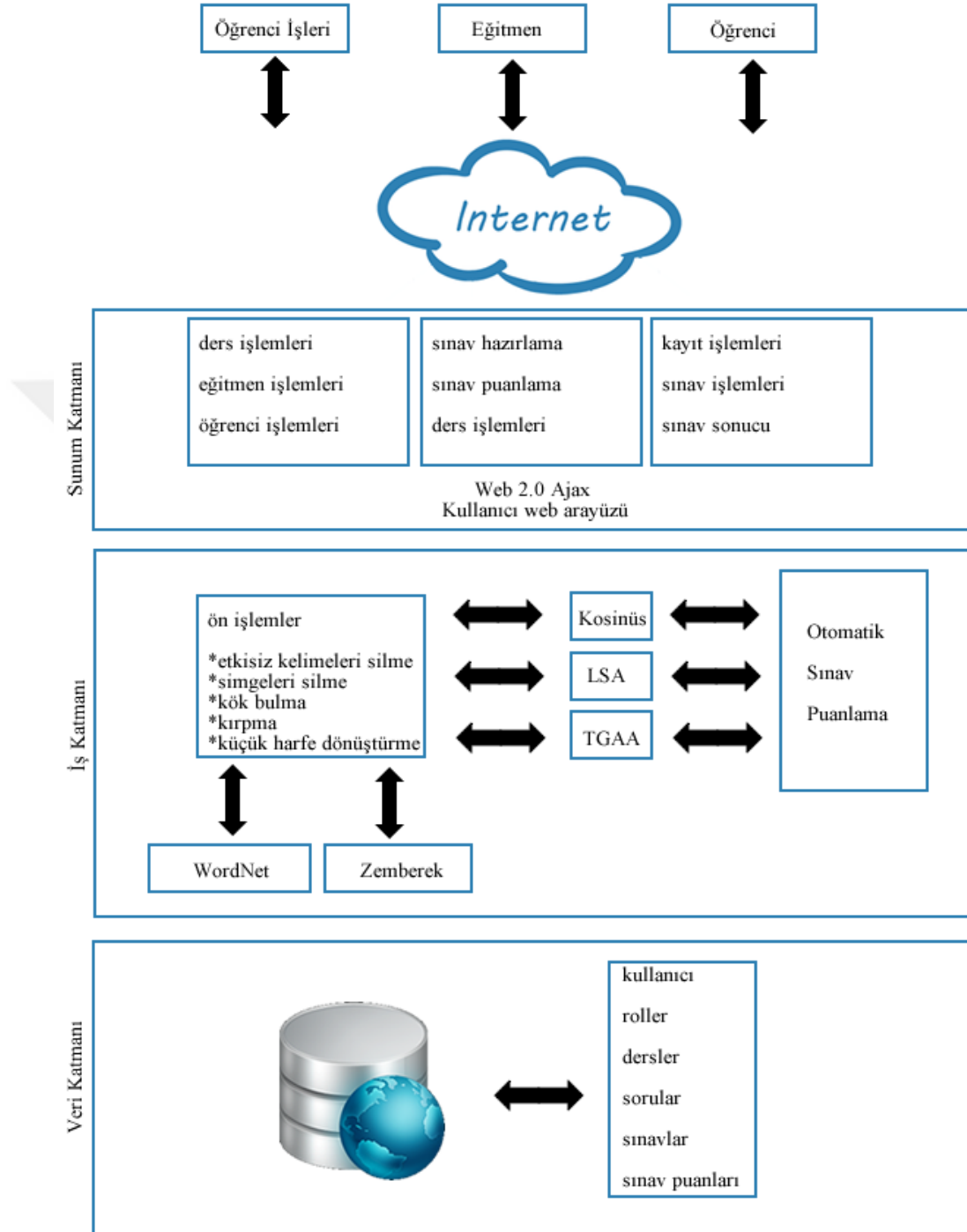
3.1. Otomatik Türkçe Sınav Puanlama Sistemi Yazılımının Tasarlanması

Otomatik Türkçe sınav puanlama sistemi (TSPS), Türkçe dilinde hazırlanan soru ve sınavları depolamak, çevrimiçi sınav yapmak ve kısa cevaplı sorulardan oluşan sınavları otomatik puanlamak için tasarlanmıştır. TSPS'in kullanıcı etkileşimli kısımları ASP.NET tabanlı web ara yüzü şeklinde gerçekleştirilmiştir. ASP.NET tabanlı web ara yüzünden girilen öğretim elemanının sınav soruları ve cevapları doğal dil işleme kodları ve yazılımda oluşturulan .NET sınıfları ile işlenerek kullanıcıya değerlendirme sisteminin test setini oluşturmuştur. Değerlendirme sisteminde öğrenci ara yüzünden girilen veriler işlenerek veri tabanına kayıt edilmiş olup, veri tabanı tasarımı MS SQL Server 2012 programıyla yapılmıştır. Yazılımın kodlanması Microsoft Visual Studio 2013 ortamında C#.NET dilinde gerçekleştirilmiştir.

Bu çalışmada tasarım tabanlı araştırma (Brown, 1992) yöntemi türevlerinden gelişimsel araştırma metodu kullanılmıştır. Öğretimsel araçların tasarımını ve öğrenmeyi konu edinen tasarım tabanlı araştırma bilginin oluşması, geliştirilmesi, kabul edilmesi ve sürekliliğinin sağlanması aşamalarını içermektedir (Collins, 1992). Gelişimsel araştırma (Richey vd., 2003); Tip 1 bir program veya ürün üzerine odaklanan ve Tip 2 araştırma sürecine odaklanan olmak üzere iki çeşitten oluşmaktadır. Bu çalışmada otomatik sınav puanlama sistemi yazılımı geliştirme sürecinde Tip 1 yöntemi kullanılmıştır.

Sistemin genel mimarisi çok katmanlı yazılım mimarisi kullanılarak geliştirilmiştir. Çok katmanlı yazılım mimarisi web ara yüzlerinin olduğu sunum katmanı, sistemin çalışmasını düzenleyen kodların olduğu iş katmanı ve veri tabanı işlemlerinden sorumlu olan veri katmanı olmak üzere üç kısımdan

oluşmaktadır (Şekil 3.1). Çok katmanlı yazılım mimarisinin kullanılması, tüm sistemin esnekliğini ve dayanıklılığını sağlamıştır.

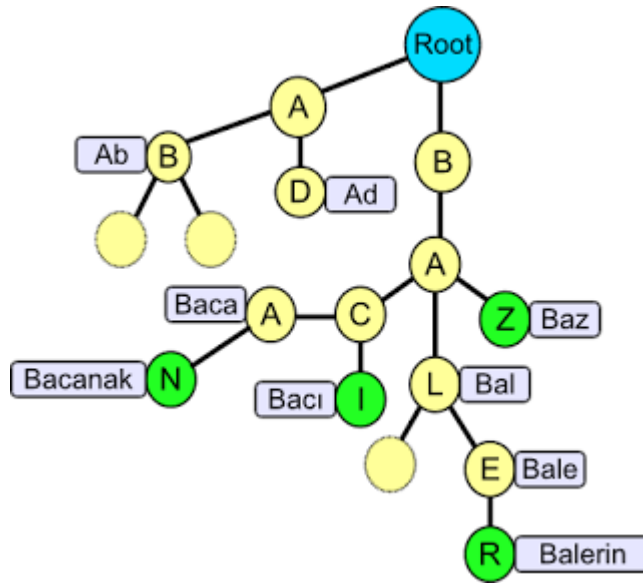


Şekil 3.1. TSPS yazılım mimarisi

Yazılımın sunum katmanında ASP.NET, AJAX teknolojileri ve C#.NET yazılım dili kullanılarak web sayfaları hazırlanmıştır. Öğretmenler bu web ara yüzlerini

kullanarak, sınav hazırlama, sınav sorularının puan değerlerini atama ve öğrenci işlerinin atadığı dersleri görüntüleme işlemlerini yapabilmektedir. Ayrıca öğretmenler sınav puanlarında ve sorularında ekleme, silme, güncelleme ve görüntüleme işlemlerini yapabilmektedir. Öğrenci işleri kullanıcısı, öğretmenlere ders atama işlemini ders işlemlerinden yapabilmektedir. Öğretmenlerin üzerindeki dersleri ve derslere kayıtlı öğrencileri görüntüleme işlemleri de öğrenci işleri kullanıcısı tarafından yapılmaktadır. Öğrenciler derse kayıt olma, sınava girme ve sınavdaki kendi sonucunu görme işlemlerini yapmaktadır.

İş katmanı Türkçe ön işlemler, WordNet, Zemberek, benzerlik motoru ve otomatik sınav puanlama modüllerinden oluşmaktadır. Ön işlemler etkisiz elemanları silme, karakter olmayan simgeleri silme, kök bulma, kırpma ve küçük harfe dönüştürme doğal dil işleme işlemlerini içermektedir. Zemberek (Akın ve Akın, 2007), Türkçe biçimbilimsel analiz yapan çatıdır, doğal dil işleme ön işlemleri bu çatıya göre yapılmaktadır. Zemberek kelime kök ağacı Şekil 3.2'de sunulmuştur. WordNet (Miller, 1986) kelimelerin anlamsal benzerlikleri için kullanılır ve bu çalışmada Türkçe WordNet kullanılmıştır.



Şekil 3.2. Zemberek kök ağacı (Akın ve Akın, 2007)

İş katmanında bulunan Türkçe ön işlemlerin yazılımda gerçekleştirilmesi için `MetinAraclari` sınıfı oluşturulmuştur. EK A'da `MetinAraclari` sınıfının kodları bulunmaktadır. Bu sınıfta gerçekleştirilen işlemler;

- Etkisiz elemanları silme işlemi için Türkçe gereksiz kelimeleri içeren `stopwords.txt` belgesi içerisindeki kelimeler silinmiştir. `Stopwords.txt` belgesi içeriği Türkçe gereksiz kelimeler EK B'de verilmiştir.
- Karakter olmayan simgeleri silme işlemi için noktalama işaretlerinin kaldırılması için `"pMetin = Regex.Replace(pMetin, @"[\w]", " ");"` kodu kullanılmıştır.
- Kök bulma işlemi için `KokBulucu` sınıfı kullanılmıştır. `KokBulucu` sınıfı Zemberek kütüphanesini kullanmaktadır ve EK C'de sunulmuştur.
- Kırpma işlemi kelimelerin önündeki ve sonundaki boşlukların yok edilmesiyle kelimelerin ayrı ayrı elde edilmesini sağlamaktadır. Kırpma işleminin gerçekleştirilmesi için `"pMetin = pMetin.Trim();"` kodu kullanılmıştır.
- Küçük harfe dönüştürme işlemi kelimelerin doğru eşleştirilmesini sağlayan bir işlemdir. Küçük harfe dönüştürme işlemi için `"pMetin = pMetin.ToLower();"` kodu kullanılmıştır.

Çalışmada eş anlamlı kelimelerin tespit edilmesi için Türk Dil Kurumunun oluşturduğu TDK Eş Anlamlar sözlüğü (Türk Dil Kurumu, 2015) kullanılmıştır. Bu sözlüğün içeriği `th_tr_TR_v2.dat` belgesinde bulunmaktadır. Bu belgenin okunması ve kelimelerin eş anlamlılarının tespiti için `EsAnlamBulucu` sınıfı oluşturulmuştur. Eş anlam bulucu sınıfı kodları EK D'de sunulmuştur. Örnek olay incelemesi için, bu sözlüğe ağ terimlerinin eş anlamlıları eklenmiştir.

Çalışmada yakın anlamlı kelimelerin tespit edilmesi için oluşturulan Türkçe WordNet, Balkanet Türkçe Wordnet (Stamou vd., 2002; Bilgin vd., 2004), Türkiye Bilişim Derneği bilişim sözlüğü (Bilişim Sözlüğü, 2015) ve Eş Anlam-Yakın Anlam sözlüğü (Türkçe Eş Anlamlar Sözlüğü, 2015) birleştirilmesi ve örnek olay çalışması için bilgisayar ağları sınavında kullanılacak terimlerin eklenmesiyle oluşturulmuştur. Bilgisayar ağları ders içeriği terimleri ve ilişkileri Çölkesen ve Örencik (2008) kitabından tez kapsamında oluşturulmuştur.

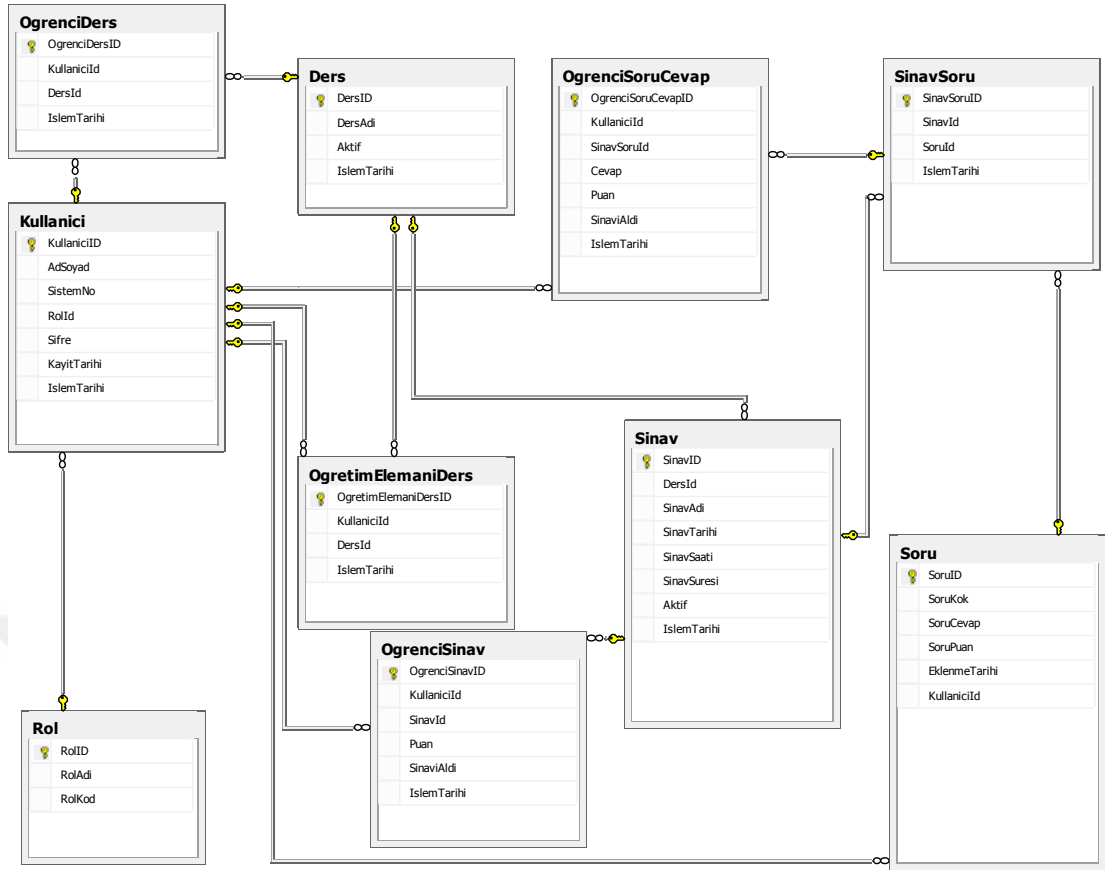
Türkçe WordNet bileşenlerinin kelime sayıları Çizelge 3.1’de sunulmuştur. Türkçe WordNet içeriği relations.txt belgesinde bulunmaktadır. Bu belgenin okunması ve kelimelerin eşanlamlılarının tespiti için YakınAnlamBulucu sınıfı oluşturulmuştur. Yakın anlam bulucu sınıfı kodları EK E’de sunulmuştur.

Çizelge 3.1. Türkçe WordNet bileşenleri

Bileşen	Geliştirici	Kelime Sayısı
Balkanet Türkçe Wordnet	Stamou vd., 2002; Bilgin vd., 2004	14.796
Eş Anlam-Yakın Anlam Sözlüğü	Türkçe Eş Anlamlar Sözlüğü, 2015	82.938
Bilgisayar Ağları Sınavı Terimleri	Tez Kapsamında Geliştirilmiştir	3442

Benzerlik motoru kosinüs benzerliği, GAA ve tümleşik gizli anlambilimsel analiz (TGAA) hesaplamalarını yapmaktadır. GAA hesaplamasında gerekli olan tekil değer ayrışımı işlemi için çapraz platform sayısal analiz ve veri işleme kütüphanesi olan ALGLIB2 (Bochkanov ve Bystritsky, 2011) kullanılmıştır. Otomatik Türkçe sınav puanlama sistemi öğrencilerin sınav sonucunu otomatik olarak belirlemektedir.

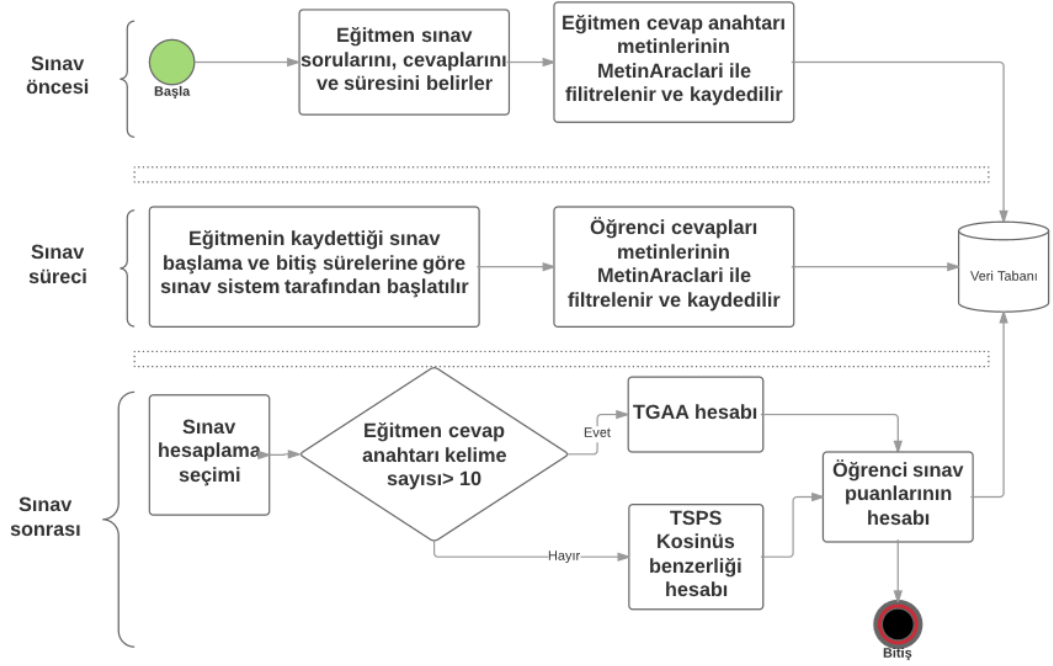
Veri katmanında TSPS yazılımının kullanıcı, roller, dersler, sorular, sınavlar ve sınav puanları gibi tüm veriler eklenmektedir. Veri katmanı sayesinde kullanıcılar ekleme, güncelleme, görüntüleme işlemlerini yapabilmektedir (Şekil 3.3).



Şekil 3.3. TSPS yazılım veri tabanı

3.2. Geliştirilen TSPS Yazılımının Hesaplama Modülü

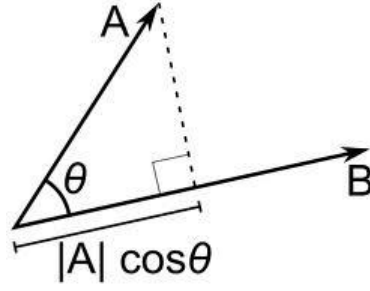
TSPS yazılımı, kosinüs benzerliği ve TGAA metodundan oluşan hibrid benzerlik ölçütü ile sınav sonuçlarını hesaplamaktadır. Öğretmenin sınav cevap anahtarındaki kelime sayısına bakarak benzerlik hesabına sistem otomatik karar vermektedir. Cevap anahtarı on kelimedenden az olma durumunda GAA hesaplaması için yeterli boyut oluşmayacağından dolayı (Landauer ve Dumais, 1997; Kontostathis, 2007), sistem otomatik olarak kosinüs benzerliğini seçmektedir. Kelime sayısı ondan fazla durumlarda ise GAA tabanlı TGAA metodu kullanılmaktadır (Şekil 3.4).



Şekil 3.4. Hesaplama yönteminin belirlenmesi

İki metin arasındaki benzerlik değerini bulmak için sıkça kullanılan yöntemlerden birisi kosinüs benzerliğidir. Kosinüs metin benzerliği iki metin arasındaki benzerliği trigonometrideki kosinüs fonksiyonu ile hesaplamaktır (Mihalcea vd., 2006). Metinler arası benzerlik hesaplanmasında öncelikle incelenen metinler vektörel olarak ifade edilmektedir. Tamamen birbirine ilişkisiz iki vektör için ise kosinüs değeri 0 olurken, tamamen aynı yönü gösteren iki vektör için kosinüs değeri 1 olacaktır. Vektör haline getirilen **A** ve **B** metinlerinin (Şekil 3.5) noktasal çarpımlarının, her iki vektör için her bir noktanın karesinin toplamının kareköklerinin birbiri ile çarpımının oranı metin benzerliği oranını vermektedir (Formül 3.1).

$$\text{Kosinüs_Benzerliği (A, B)} = \frac{\text{dot(A, B)}}{||\mathbf{A}|| \cdot ||\mathbf{B}||} \quad (3.1)$$



Şekil 3.5. Kosinüs benzerliği vektörel gösterimi (Skowron ve Suraj, 2012).

Örneğin;

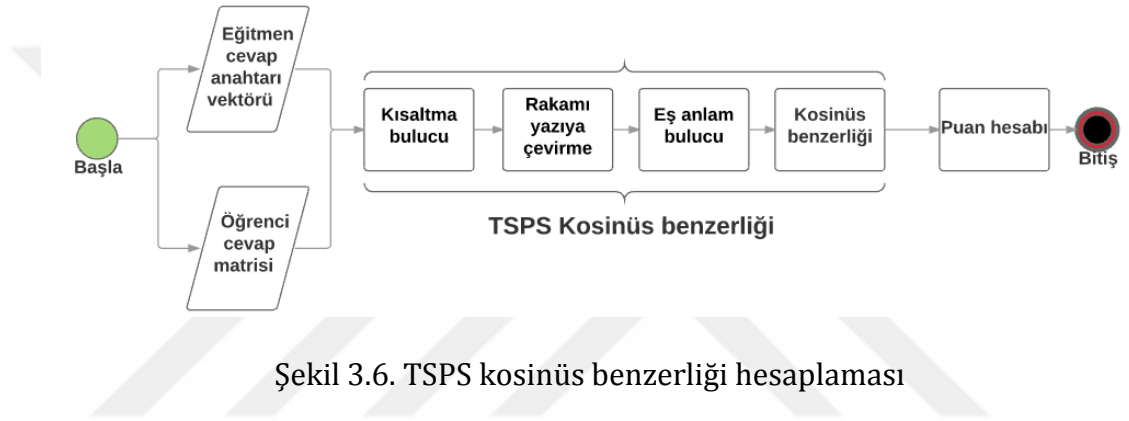
Metin 1 : Hub fiziksel katmanda çalışır

Metin 2 : Fiziksel katmanda yineleyici, Ethernet çalışır

Bu iki metinden terim frekansına göre özellik çıkarımı yapılması için metinlerdeki tüm terimler (Hub, fiziksel, katmanda, yineleyici, Ethernet, çalışır) listelenir. Elde edilen terimlerin dokümanlarda kaçar kere geçtiği vektör uzay modeli ile vektör şeklinde gösterildiğinde, Metin1= [1,1,1,0,0,1] ve Metin2=[0,1,1,1,1,1] vektörleri oluşturulmuştur. Burada iki vektör arasındaki kosinüs bağlantısı için iki vektörün skaler çarpımının iki vektörün vektörel çarpımına oranı alınmıştır. Çarpım için, boyutların ayrı ayrı çarpılmasından elde edilen değerlerin toplamı hesaplanmıştır. Burada iki vektör arasındaki kosinüs bağlantısı için iki vektörün skaler çarpımının iki vektörün vektörel çarpımına oranı alınmıştır. $\text{dot}(\text{Metin1}, \text{Metin2}) = [1,1,1,0,0,1] \cdot [0,1,1,1,1,1] = (1) \cdot (0) + (1) \cdot (1) + (1) \cdot (1) + (0) \cdot (1) + (0) \cdot (1) + (1) \cdot (1) = 3$ sonucu elde edilmiştir. Bu sonuç iki vektörün noktasal çarpımıdır. Vektörlerin boyutlarını hesaplanması için öklit uzayından faydalanılır ve her boyuttaki değerlerinin karelerini toplayarak toplamın karekökü alınmıştır. $\|\text{Metin1}\| = \sqrt{(1)^2 + (1)^2 + (1)^2 + (0)^2 + (0)^2 + (1)^2} = 2$ ve $\|\text{Metin2}\| = \sqrt{(0)^2 + (1)^2 + (1)^2 + (1)^2 + (1)^2 + (1)^2} = 2.2360679775$ olarak boyutları da hesaplanmış oluyor. Formülde elde edilen değerleri yerine konulduğunda; $\cos(\text{Metin1}, \text{Metin2}) = \frac{\text{dot}(\text{Metin1}, \text{Metin2})}{(\|\text{Metin1}\| \|\text{Metin2}\|)} = \frac{3}{(2 \times 2.2360679775)} = \frac{3}{4.472135955} = 0.67082039325$ olarak iki doküman arasındaki kosinüs

benzerliđi bulunmuştur. Sonuç olarak, bu iki metnin kosinüs benzerliđine göre %67'si benzer olduđu tespit edilmiştir.

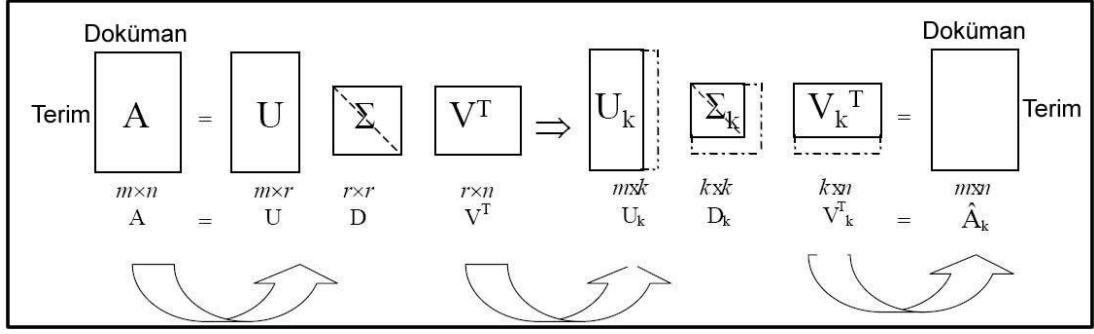
Kosinüs benzerliđinin hesaplanması için yazılımda Cosine sınıfı oluşturulmuştur. Cosine sınıfında eđitmen cevabındaki kelimenin eş anlamlı listesinde öğrencinin cevabı varsa, öğrenci cevabı eđitmenin kelimesine eşitlenerek işlem yapılmıştır. Bu durum öğrenci ve eđitmenin eş anlamlı kelimeler kullanma durumunda, kelimelerin birbirine eşlenmesini sağlamaktadır (Şekil 3.6). Cosine sınıfı kodları EK F'de sunulmuştur.



Şekil 3.6. TSPS kosinüs benzerliđi hesaplaması

Gizli anlambilimsel analiz (Landauer vd., 1998) ile metin benzerliđi hesaplanmasında, metinde geçen her kelime satırda terim ve her doküman sütunda olmak üzere **A** matrisi oluşturulmaktadır. Şekil 3.7'de terim sayısı m , doküman sayısı n , kavram r ve kavram köşegen matrisi Σ ile ifade edilmektedir. Oluşturulan matrise tekil deđer ayrışımı uygulanmaktadır (Formül 3.2). Tekil deđer ayrışımı sonucu oluşan **U** matrisi dikey, **D** matrisi çapraz ve **V** matrisi ise yatay bilgilerini tutmaktadır (Şekil 3.7). Elde edilen tüm matrisler k boyuta azaltılarak \mathbf{U}_k , \mathbf{D}_k ve \mathbf{V}_k^T matrisleri elde edilmektedir. $\hat{\mathbf{A}}_k$ matrisi doküman terim matrisinin k boyutta azaltılmış halini ifade etmektedir.

$$\mathbf{A} = \mathbf{UDV}^T \quad (3.2)$$



Şekil 3.7. Tekil değer ayrışımı (Landauer vd., 1998)

Gizli anlambilimsel analiz, terim ve doküman bilgileri içeren önceden oluşturulmuş matrisden elde edilen bilgilerle, dışarıdan yeni bir doküman eklendiğinde, eklenen yeni dokümanla eski dokümanların arasındaki metin benzerliğinin hesaplandığı bir yöntemdir. Örneğin, D_1, \dots, D_{10} olmak üzere 10 farklı dokümanın içerdiği terimleri aşağıda listemiştir. Bu dokümanlarla frekansa göre oluşturulan terim/doküman matrisi Şekil 3.8'de sunulmuştur. Terim/doküman matrisi A olarak ifade edilmiştir.

D1= repeater, kablo, mau, alıcı, verici, konnektör, elektriksel, iletim

D2= hub, repeater, kablo

D3= mau, alıcı, verici, konnektör, elektriksel, iletim

D4= hub, nic, alıcı, verici, elektriksel

D5= hub, repeater, kablo, mau, konnektör, elektriksel, iletim

D6= hub, nic, alıcı, verici

D7= hub, repeater

D8= hub, repeater, nic, alıcı, verici, konnektör

D9= hub, repeater, nic, alıcı, verici

D10= hub, kablo, nic, mau, verici, konnektör, iletim

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
hub	0	1	0	1	1	1	1	1	1	1
repeater	1	1	0	0	1	0	1	1	1	0
kablo	1	1	0	0	1	0	0	0	0	1
nic	0	0	0	1	0	1	0	1	1	1
mau	1	0	1	0	1	0	0	0	0	1
alıcı	1	0	1	1	0	1	0	1	1	0
verici	1	0	1	1	0	1	0	1	1	1
konnektör	1	0	1	0	1	0	0	1	0	1
elektriksel	1	0	1	1	1	0	0	0	0	0
iletim	1	0	1	0	1	0	0	0	0	1

Şekil 3.8. Gizli anlambilimsel analiz terim/doküman matrisi

A matrisi için tekil değer ayrışımı elde edilen matrisler k boyutunda eksiltme işlemi gerçekleştirildiğinde \mathbf{U}_k , \mathbf{D}_k ve \mathbf{V}_k^T matrisleri oluşmuştur (Şekil 3.9).

$$\mathbf{U}_k = \begin{bmatrix} -0.4001 & -0.3376 \\ -0.3023 & 0.0511 \\ -0.2307 & 0.2841 \\ -0.2906 & -0.4597 \\ -0.2589 & 0.3936 \\ -0.4203 & -0.2048 \\ -0.4203 & -0.2048 \\ -0.2530 & 0.3172 \\ -0.2434 & 0.3110 \\ -0.2589 & 0.3936 \end{bmatrix}$$

$$\mathbf{D}_k = \begin{bmatrix} 5.7196 & 0 \\ 0 & 3.0345 \end{bmatrix}$$

$$\mathbf{V}_k^T = \begin{bmatrix} -0,4175 & -0,1631 & -0,3243 & -0,3103 & -0,3405 & -0,2677 & -0,1228 & -0,3648 & -0,3206 & -0,3986 \\ 0,4419 & -0,0007 & 0,3314 & -0,2952 & 0,4657 & -0,3977 & -0,0944 & -0,2763 & -0,3809 & -0,0447 \end{bmatrix}$$

Şekil 3.9. Tekil değer ayrışımı sonuçları

D11 vektörünün **A** matrisinden elde edilen verilere göre, \mathbf{D}_1 \mathbf{D}_{10} dokümanlarına metin benzerliğinin hesaplanması için Formül 3.3 uygulanır.

$$\mathbf{D11} = \mathbf{D11}^T \cdot \mathbf{U}_k \cdot \mathbf{D}_k^{-1} \quad (3.3)$$

Formül hesaplandıktan sonra $\mathbf{D11} = (-0.5383, 0.1792)$ elde edilmiştir. \mathbf{V}_k^T ile $\mathbf{D11}$ arasında kosinüs benzerliği uygulanarak $\mathbf{D11}$ dokümanın, diğer dokümanlara metin benzerliği hesaplanmaktadır. Formül 3.1'den elde edilen sonuçlara göre $\mathbf{D11}$ dokümanın diğer dokümanlara olan benzerliği $\mathbf{D1} > \mathbf{D5} > \mathbf{D3} > \mathbf{D10} > \mathbf{D8} > \mathbf{D9} > \mathbf{D4} > \mathbf{D6} > \mathbf{D2} > \mathbf{D7}$ şeklinde bulunmuştur.

TSPS yazılımı, TGAA metodu kullanarak sınav sonuçlarını otomatik olarak hesaplamaktadır. TGAA metodunda GAA metodundan farklı olarak eşanlam, alt kavram, kısaltma ve rakam durumları vektör oluşturulmadan önce tespit etmektedir. Aynı soru cevabı için oluşan kelime vektörleri Şekil 3.10'de GAA ve Şekil 3.11'de TGAA için sunulmuştur. Örnek olay çalışması sınavında bulunan, "OSI referans modelinin birinci katmanındaki cihazları yazınız" sorusuna verilen cevap vektörleri kelime ön işlemcisinden geçtikten sonra doğrudan GAA vektörü oluşturulduğunda, Şekil 3.10'daki 15 farklı kelime elde edilmiştir. Fakat aynı cevaplar eşanlam, alt kavram, kısaltma ve rakam durumları incelenerek, yani TGAA metodu kullanılarak incelendiğinde Şekil 3.11'deki gibi 10 kelime elde edilmiştir. Bu iki kelime vektörü arasındaki fark incelendiğinde "yineleyici" kökü olan "yinele" ile "repeater" kelimesi aynı cihazı ifade ettiği için TGAA'da sadece "repeater" olarak kelime vektörüne eklenmiştir. Böylece öğrencinin cevabında anlatmak istediği kelime ile eğitmenin cevap anahtarındaki kelime eşleştirilmiştir. Bu eşleme, TGAA benzerliğinde öğrencinin doğru cevabının hak ettiği puanı almasını sağlamaktadır.

```

ogrenciKisiselCevapListesi.Add(CevapAnahtari);
foreach (string[] cevaplar in ogrenciKisiselCevapListesi)
{
    foreach (var item in cevaplar)
    {
        if (!KelimelerSatiri.Contains(item))
            KelimelerSatiri.Add(item);
    }
}
ogrenciKisiselCevapListesi.Sort();
KelimelerSatiri.Sort();
int M = KelimelerSatiri.Count;
int N = pOgrenciCevapListesi.Count - 1);

List<int[]> VektorListesi = new List<int[]>(N);
foreach (var item in ogrenciKisiselCevapListesi)
{
    int[] vektor = new int[M];
    for (int i = 0; i < M; i++)
    {
        vektor[i] = item[i].Count;
    }
}

```

Index	Term
[0]	"alıcı"
[1]	"bir"
[2]	"card"
[3]	"cihaz"
[4]	"ethernet"
[5]	"hub"
[6]	"interface"
[7]	"kablo"
[8]	"katman"
[9]	"network"
[10]	"nic"
[11]	"repeater"
[12]	"voltaj"
[13]	"yinele"
[14]	"yükselti"

Şekil 3.10. Örnek GAA kelime vektörü

```

ogrenciKisiselCevapListesi.Add(CevapAnahtari);
foreach (string[] cevaplar in ogrenciKisiselCevapListesi)
{
    foreach (string item in cevaplar)
    {
        string[] icdizi = item.Split(' ');
        foreach (var ic in icdizi)
        {
            if (!KelimelerSatiri.Contains(ic) && ic[0] != '|')
                KelimelerSatiri.Add(ic);
        }
    }
}
ogrenciKisiselCevapListesi.Sort();
KelimelerSatiri.Sort();
int M = KelimelerSatiri.Count;
int N = pOgrenciCevapListesi.Count - 1);

```

Index	Term
[0]	"alıcı"
[1]	"bir"
[2]	"card"
[3]	"cihaz"
[4]	"hub"
[5]	"interface"
[6]	"kablo"
[7]	"katman"
[8]	"network"
[9]	"repeater"

Şekil 3.11. Örnek TGAA kelime vektörü

Geliştirilen TGAA metodu, eğitimci cevap anahtarındaki kelimeleri inceleyip, öğrenci cevabındaki eş anlam, alt kavram, kısaltma ve rakam durumlarını tespit etmektedir. Vektör uzay modeli ile eğitimci cevap anahtarı kelimeleri ve öğrenci cevabı (ogrenciCevapListesi) olmak üzere dizi şeklinde ifade edilmiştir. Ayrıca eğitimci cevap anahtarındaki kelimelerin eş anlamlıları (EsAnlamliListesi) ve yakın anlamlıları (YakinAnlamliListesi) dizi şeklinde tutulmuştur. Metinler arası anlamsal benzerlik tespiti için aşağıdaki kurallar uygulanmaktadır. TGAA metodu kuralları;

- Eğer eğitimci cevap anahtarından elde edilen EsAnlamliListesi içeriğindeki bir kelime ogrenciCevapListesi'nde bulunuyorsa, EsAnlamListesi'ni oluşturan cevap kelimesi ogrenciCevapListesi'deki kelimeye kopyalanır. Bu durum aynı anlamlı kelimelerin vektör uzay modelinde tekrar yeni bir terim oluşturmaya gerek kalmadan eşlenmesini sağlamaktadır. Yeni durumda öğrenci cevabında, eğitimci cevap anahtarındaki bir kelimenin eş anlamlısı varsa, öğrenci cevap anahtarındaki kelime silinir ve eğitimci cevap anahtarındaki kelime öğrenci cevabına kopyalanır (Şekil 3.12). Eş anlam bulucu sınıfı EK D'de sunulmuştur.

```

Girdiler: cevapAnahtarıKelimeleri  $ca_1, \dots, ca_m$  ,
ogrenciCevapListesi  $oclist_1, \dots, oclist_n$  ,
EsAnlamliListesi  $ealist_1, \dots, ealist_p$  :
for i=1 to m do
    for j=1 to n do
        for k=1 to p do
            if((oclist(j,k))  $\in$  ealist( $ca_i$ ))
                oclist(j,k)=  $ca_i$ ;
            end;
        end;
    end;
end;

```

Şekil 3.12. Eş anlam bulucu algoritması

- Eğer eğitimci cevap anahtarı ve öğrenci cevabı kelimeleri alt anlam ilişkisine sahipse yakın anlam işlemi yapılır (Şekil 3.13). Kelimelerin alt

anlam ilişkisi WordNet'deki düğüm uzaklığı dikkate alınarak μ katsayısı hesaplanır. YakınAnlamliListesi içeriğindeki bir kelime ve ogrenciCevapListesi'nde bulunuyorsa, YakınAnlamListesi'ni oluşturan cevap kelimesi ogrenciCevapListesi'deki kelimeye μ katsayısı ile eklenir (Şekil 3.14). Sonrasında öğrenci cevabında tespit edilen yakın anlamlı kelimeler listelenir ve eğitmen cevap anahtarındaki ilgili kelimenin frekansına eklenir. Kelimeler arası yakın anlamlar Türkçe WordNet ile tespit edilmiştir. Ayrıca ogrenciCevapListesi'nde iki veya daha fazla alt anlam varsa, aynı şekilde μ katsayısı hesaplanmaktadır. Şekil 3.14'de Türkçe WordNet düğümü örneği olarak "Bilgisayar Ağı" verilmiştir. "Bilgisayar Ağı" ile OSI arasında bir düğüm uzaklık vardır. Yine "Bilgisayar Ağı" ile "Fiziksel" arasındaki düğüm uzaklığı incelenirse, iki düğüm uzaklığıdır. Bu çalışmada bir düğüm uzaklığı dikkate alınmış olup, μ katsayısı 0.5 olarak alınmıştır. Eğer iki düğüm uzaklığı dikkate alınsaydı, iki düğüm uzaklıktaki yakın anlamlı kelimeler için μ katsayısı 0.25 olarak alınacaktı (Formül 3.4). Formül 3.4'deki uzaklık değişkeni iki düğüm arasındaki uzaklığı ifade etmektedir. Yakın anlam bulucu sınıfı EK E'de sunulmuştur.

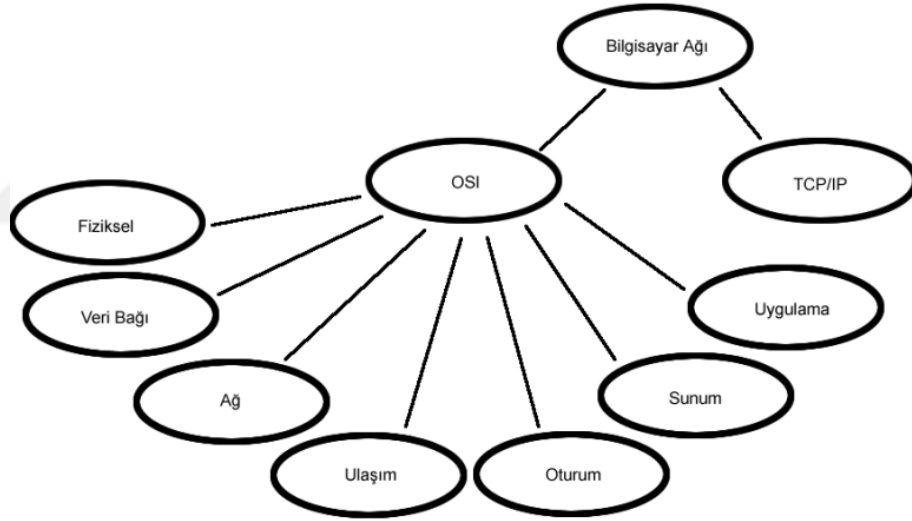
$$\mu = \frac{1}{2^{\text{uzaklık}}} \quad (3.4)$$

```

Girdiler: cevapAnahtarıKelimeleri  $ca_1, \dots, ca_m$ ,
ogrenciCevapListesi  $oclist_1, \dots, oclist_n$ ,
YakinAnlamliListesi  $yalist_1, \dots, yalist_p$  :
for i=1 to m do
  for j=1 to n do
    for k=1 to p do
      if((oclist(j,k))  $\in$  yalist( $ca_i$ ))
        oclist(j,k) = "|" +  $ca_i$ ;
      end;
    end;
  end;
end;

```

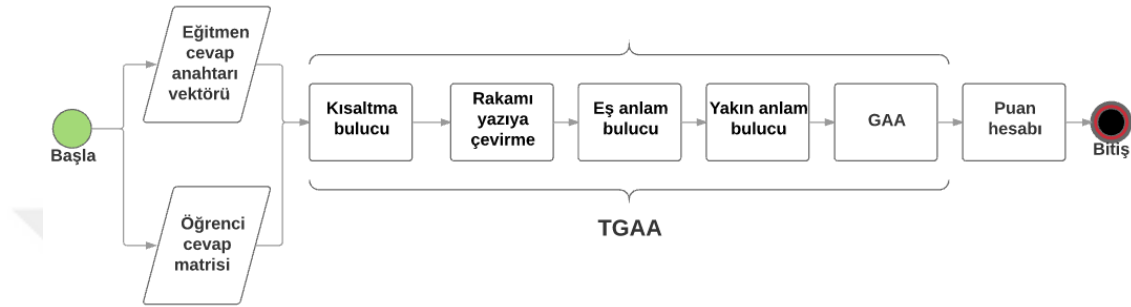
Şekil 3.13. Yakın anlam bulucu algoritması



Şekil 3.14. Türkçe WordNet düğümleri

- Eğer öğretmen cevap anahtarı veya öğrenci cevabı (ogrenciCevapListesi) kelimeleri kısaltma içeriyorsa, her iki vektör için de kısaltmalar uzun hali ile değiştirilmektedir. Örneğin “SDÜ” kısaltması “Süleyman Demirel Üniversitesi” olarak değiştirilmektedir. Kısaltmaların uzun durumlarına dönüştürülmesi için KısaltmaBulucu sınıfı oluşturulmuştur. Kısaltma bulucu sınıfı kodları EK G’de sunulmuştur.

- Eğer eğitmen cevap anahtarı veya öğrenci cevabı (ogrenciCevapListesi) kelimeleri vektörleri rakam içeriyorsa, her iki vektör için de rakamlar yazı hali ile değiştirilmektedir. Örneğin “6” kısaltması “altı” olarak değiştirilmektedir. Rakamları yazıya çevrilmesi için RakamiYaziyaCevir sınıfı oluşturulmuştur. EK H’de rakamların yazıya çevrilmesi kodları yer almaktadır.



Şekil 3.15. TGAA hesaplaması

Kurallar gerçekleştirildikten sonra TGAA hesaplamasında (Şekil 3.15), GAA hesabı yapılmaktadır. GAA hesabını sonucunda oluşan vektörler kosinüs benzerliği ile ölçülmektedir. GAA hesaplamasının fonksiyonu EK I’de ve TGAA hesaplamasının fonksiyonu EK İ’da sunulmuştur. TGAA gizli anlambilimsel çekirdek (Cristianini vd., 2002) hesaplamasını kullanmamaktadır. Bunun yerine, ön işlemlerde direkt olarak yukarıdaki belirtilen eş anlam, alt anlam kuralları ve formülleri uygulanmaktadır.

3.3. TGAA ile Gizli Anlambilimsel Çekirdek Yönteminin Karşılaştırılması

Gizli anlambilimsel çekirdek (Cristianini vd., 2002) yöntemi ile metinlerarası benzerlik hesaplaması yapılmaktadır. Kim ve Kim (2008) çalışmalarında WordNet yapısını çekirdek yöntemine uyarlayarak otomatik değerlendirme sisteminde kullanmışlardır. Araştırmada öncelikle, WordNet hiyerarşik yapısını dönüştürmek için, terim-eş küme matrisini oluşturulmuştur. Terim-eş küme matrisi, Formül 3.5’deki şekliyle metinden elde edilen terimler t_i ve N adet eş küme kelimeleri $s_1, s_2, \dots, s_i, \dots, s_N$ kullanılarak oluşturulmuştur. Eş küme kelimeleri WordNet hiyerarşisine göre eş anlamlı kelimeler ise 2 değerini,

1 uzaklıklı kelimeler ise 1 değerini ve 2 uzaklıklı kelimeler ise 0.5 değerini almıştır. Eş küme oluşturulurken 0, 1 ve 2 uzaklıklı kelimeler dikkate alınmış olup, 2'den büyük uzaklıktaki kelimeler kullanılmamıştır.

$$t_i = \langle s_1, s_2, \dots, s_i, \dots, s_N \rangle \quad (3.5)$$

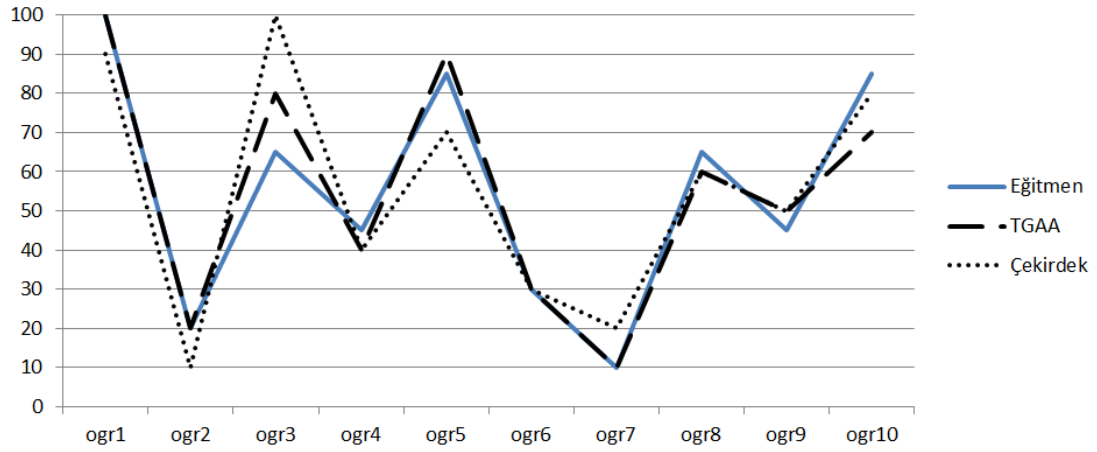
Formül 3.5'e oluşturulan terim-eş küme matrisi **A** Formül 3.2'deki gibi tekil değer ayrışımına tabi tutulmuş ve elde edilen değerler k boyutuna eksiltildiğinde elde edilen \mathbf{U}_k matrisi **P** matrisi olarak atanmıştır. İki metin arası benzerlik hesabı Formül 3.6'e göre hesaplanmıştır.

$$\text{benzerlik}(\mathbf{d}_1, \mathbf{d}_2) = \cos(\mathbf{P}^T \mathbf{d}_1, \mathbf{P}^T \mathbf{d}_2) = \frac{\mathbf{d}_1^T \mathbf{P} \mathbf{P}^T \mathbf{d}_2}{\|\mathbf{P}^T \mathbf{d}_1\| \|\mathbf{P}^T \mathbf{d}_2\|} \quad (3.6)$$

Anlambilimsel analiz yapan TGAA ile gizli anlambilimsel çekirdek yönteminin karşılaştırılmasını gerçekleştirmek üzere, 10 öğrencinin katıldığı bir sınav oluşturularak, TGAA ve gizli anlambilimsel çekirdek yöntemi ile puan hesabı yapılmıştır. Aynı zamanda sınav eğitmen tarafından da değerlendirilmiştir. Çizelge 3.2'de eğitmen, TGAA ve gizli anlambilimsel çekirdek puanları verilmiştir.

Çizelge 3.2 Eğitmen, TGAA ve gizli anlambilimsel çekirdek puanları

	Eğitmen	TGAA	Gizli Anlambilimsel Çekirdek
ogr1	100	100	90
ogr2	20	20	10
ogr3	65	80	100
ogr4	45	40	40
ogr5	85	90	70
ogr6	30	30	30
ogr7	10	10	20
ogr8	65	60	60
ogr9	45	50	50
ogr10	85	70	80



Şekil 3.16. Öğretmen, TGAA ve gizli anlambilimsel çekirdek puanları

Elde edilen verilere göre öğretmenin verdiği puanlara TGAA puanlarının, gizli anlambilimsel çekirdek puanlarına göre daha yakın olduğu tespit edilmiştir (Şekil 3.16). Puanlararası Pearson korelasyonu sonuçları Çizelge 3.3'de verilmiştir.

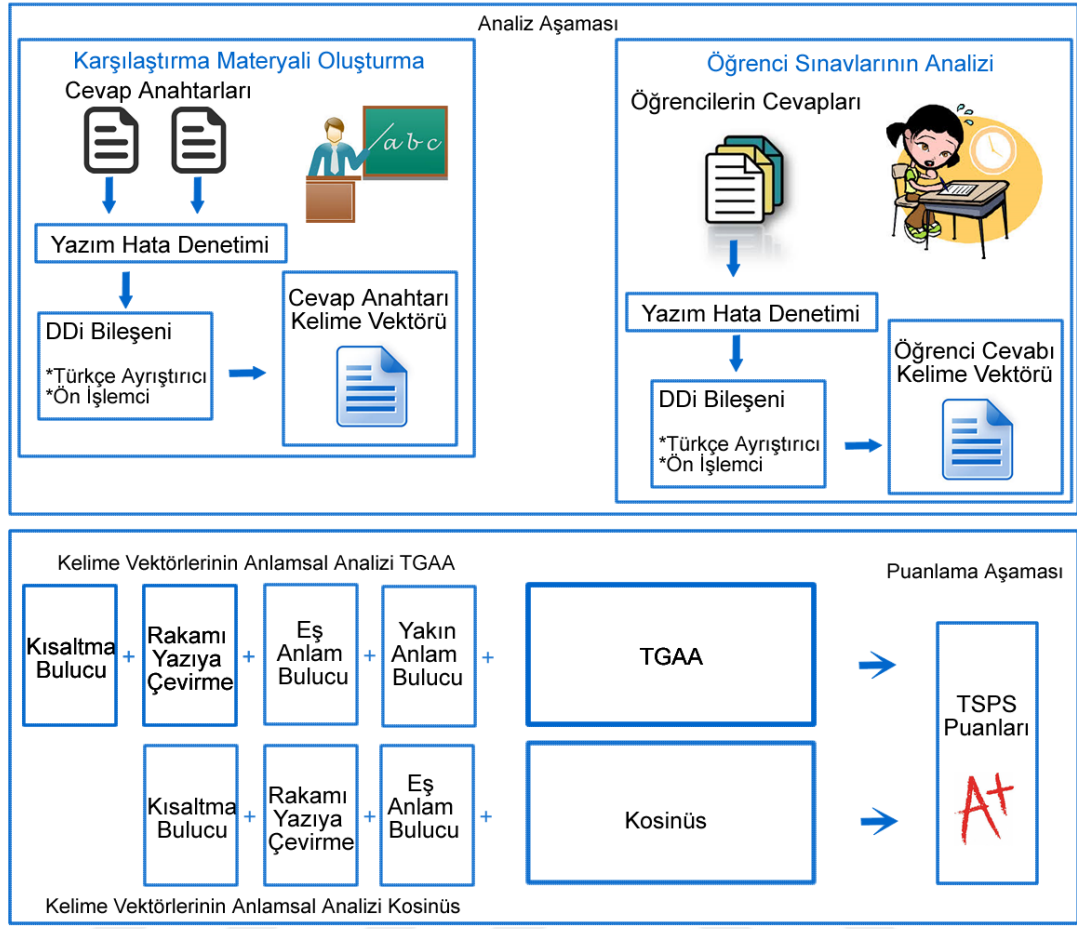
Çizelge 3.3. Puanlararası Pearson korelasyonu sonuçları

		TGAA	Öğretmen	Gizli Anlambilimsel Çekirdek
TGAA	Pearson Correlation	1	,966(**)	,927(**)
	Sig. (2-tailed)		,000	,000
	N	10	10	10
Öğretmen	Pearson Correlation	,966(**)	1	,887(**)
	Sig. (2-tailed)	,000		,001
	N	10	10	10
Gizli Anlambilimsel Çekirdek	Pearson Correlation	,927(**)	,887(**)	1
	Sig. (2-tailed)	,000	,001	
	N	10	10	10

3.4. TSPS Yazılımının Deęerlendirme Yöntemi

Bu çalışmada; gelişimsel araştırma modeli (Richey vd., 2003) olan, Tip 2 çeşidi araştırma sürecine odaklanmaktadır. Bu çalışmada geliştirilen TSPS yazılımının değerlendirme sürecinde Tip 2 yöntemi kullanılmıştır. Çalışma uzaktan eğitimde kullanılan öğretim yönetim sistemlerinden kullanılmak üzere otomatik sınav puanlama sistemi yazılımının geliştirilmesi ve yazılımın test edilmesi süreçlerini kapsamaktadır.

TSPS yazılımı, Süleyman Demirel Üniversitesi Bilgisayar Mühendisliği Bölümü'nde çevrimiçi sınav olarak uygulanmıştır. Yazılımının test edilmesi amacıyla, bilgisayar ağları dersi için on soruluk bir deneme sınavı hazırlanmıştır (EK J). Sorulara iki eğitmen tarafından iki farklı cevap anahtarı oluşturulmuş, analiz aşaması için karşılaştırma materyalleri elde edilmiştir (Şekil 3.17). Deneme sınavı 2014-2015 Eğitim Öğretim yılında, Bilgisayar Mühendisliği Bölümü'nde eğitim gören kırk bir öğrenci ile gerçekleştirilmiş, öğrenci sınavlarının analizi ve puanlaması TSPS yazılımı tarafından yapılmıştır. Deneme sınavının derse etki etmeyeceği ve bir araştırma projesi kapsamında olduğu öğrencilere önceden anlatılmıştır. Sınavların otomatik puanlanması aşamasında, TGAA metodu ve kosinüs benzerliği kullanılmıştır.



Şekil 3.17. TSPS analiz ve puanlama aşamaları

Gerçekleştirilen sınav iki alan uzmanı eğitimden tarafından birbirinden bağımsız olarak değerlendirilmiş ve puanlar Eğitimci1 Puan ve Eğitimci2 Puan değişkenleri olarak kaydedilmiştir. Aynı eğitimciler TSPS için farklı cevap anahtarları hazırlamış olup, TSPS her iki cevap anahtarı için değerlendirme yapmıştır. Buradan elde edilen puanlar da TSPS puanı olarak kaydedilmiştir.

Bilgisayar yazılımının performansını belirlemek amacıyla öncelikle eğitimci ve bilgisayar değerlendirmeleri arasındaki uyumu hesaplamak için değişkenlerin arasındaki Kendall W hesaplanmıştır. Kendall uyumluluk katsayısı; sıralı (ordinal) ölçekte değerlendiriciler arası uyumluluğu değerlendirmek için kullanılır. Kendall W; 0 (uyumluluk yok) ile 1 (tam uyumluluk) aralığındadır (Kendall ve Babington-Smith, 1939). Bu hesaplamada birinci ve ikinci cevap anahtarına göre hesaplanan TSPS puanlarının aritmetik ortalaması alınarak TSPSORT değişkeni de dikkate alınarak en iyi uyum aranmıştır. Ayrıca

oluşturulan tüm değişkenler arasındaki değişimler Pearson korelasyon katsayılarıyla hesaplanmış ve puanların ortalamaları arasında anlamlı farklılıkların olup olmadığı bağımlı gruplar t-testi ile irdelenmiştir.

Şekil 3.17'nin analiz aşamasında verilen kelimenin hata denetimi işlemi kelimelerin hata denetimini gerçekleştirmenin yanı sıra, hatalı kelimeler için öneri sunmaktadır. Hata denetimi işlemleri için HataDenetemi sınıfı oluşturulmuş olup, bu sınıfın kodları EK K'de sunulmuştur.

Hata denetimi işlemi için Türkçe kelimelerin köklerini külliyatta tutan, eklerin dizilimi kurallarını dikkate alarak ekler için çizge yapısını oluşturan Zemberek yazılımı kullanılmıştır. Kelimenin doğru veya yanlış yazılması bu yapının kelimeDenetle fonksiyonuyla gerçekleştirilmiştir.

Kelimenin hata denetimi yapıldıktan sonra yanlış yazıldığı tespit edilen kelimelerin düzeltilmesi için kelime önerileri oluşturulmuştur. Yazım yanlış yapılan kelime önerisinin oluşturulması aşamasında n-gram yöntemi kullanılmıştır. N-gram verilen bir dizilimdeki tekrar oranını bulmaya yarayan yöntemdir. N tekrarın kontrol edildiği değer olup gram ise bu tekrarın dizilim içerisindeki ağırlığını ifade etmektedir. N-gram metin benzerliğini ilgilenilen metin içerisinde belirlenen n tekrarındaki ağırlıklara göre hesaplanmasını sağlamaktadır (Damashek,1995).

Yazılımda yanlış yazılan kelimenin uzunluğu miktarında n-gramları çıkartılmış, elde edilen n-gramlar arasında K eşik değeri kullanılarak seçim yapılmıştır. K eşik değerinden sonraki tüm n-gramlar sözlükte bulunma durumuna göre bir değişiklik yapılarak test edilmiştir. Bu çalışmadaki örnek yazılımda K eşik değeri 3 olarak test edilmiştir. Elde edilen öneriler Levenstein düzenleme uzaklık algoritması ile elenerek en yakın 2 uzaklığa göre öneri listesine eklenmiştir. Elemeden sonra öneri sıralaması önerilen kelimenin frekansına göre belirlenmiştir. Frekansı en fazla olan öneri kelime ilk önerilen kelime olarak atanmıştır. Levenstein düzenleme uzaklık algoritmasının yazılımda

kullanılması için Levenstein sınıfı oluşturulmuş olup EK L'de yazılım kodları sunulmuştur.



4. ARAŞTIRMA BULGULARI

Bu bölümde, TSPS yazılımının kullanıcı ara yüzleri ve TSPS yazılımının başarı testi bilgilerine yer verilmiştir.

4.1. TSPS Yazılımının Kullanıcı Arayüzleri

TSPS yazılımı Türkçe otomatik sınav puanlama işlemi yapmak amacıyla geliştirilmiştir. TSPS yazılımı ara yüzü, kullanıcı dostu web 2.0 sayfalardan oluşmakta olup, giriş ekranı Şekil 4.1’de verilmiştir.



Şekil 4.1. TSPS yazılımı giriş ekranı

Eğitmenler bu web ara yüzlerini kullanarak, sınav hazırlama, sınav sorularının puan değerlerini atama ve öğrenci işlerinin atadığı dersleri görüntüleme işlemlerini yapabilmektedir. Ayrıca eğitmenler sınav puanlarında ve sorularında ekleme, silme, güncelleme ve görüntüleme işlemlerini yapabilmektedir. Eğitmenler için, sınav hazırlama ekran görüntüsü Şekil 4.2’de ve otomatik puanlama sonuçları görüntüsü Şekil 4.3’de verilmiştir.

Merhaba Ebru Yılmaz İNCE Çıkış

OTOMATİK DEĞERLENDİRME SİSTEMİ

SULTAN SAİD İLİTİM S. D. 1997

Dersler Sinavlar

SINAV SORU LİSTESİ

Ders Adı: Ağ Temelleri

Sınav Adı: yeni deneme

Sınav Tarihi: 17.5.2015

Sınav Saati: 17:35

Sınav Süresi (dk): 60

Aktif:

Sınava ait 10 adet soru bulunmaktadır. [Yeni Soru Ekle](#)

Soru No	Soru	Cevap	Puan	
1	OSI referans modelinin birince katmanındaki cihazları yazınız	network interface card yineleyici hub multi access unit kablo alıcı ve verici	10	Düzenle Sil
2	Ağ topoloji türlerini yazınız	doğrusal topoloji halka topoloji yıldız topoloji ağaç topoloji karmaşık topoloji	10	Düzenle Sil
3	OSI referans modelindeki veri iletim	ağ katmanından aldığı veri paketlerine hata kontrol bitlerini ekleyerek çerçeve frame halinde fiziksel katmana iletme işinden sorumludur iletilen çerçevenin doğru mu yoksa yanlış mı iletildiğini kontrol eder eğer çerçeve hatalı iletilmişse çerçevenin yeniden gönderilmesini sağlar ayrıca ağ üzerindeki	10	Düzenle

Şekil 4.2. Eğitimci için sınav hazırlama ekran görüntüsü

Sınav No: 15

Sınav Adı: yeni deneme

Sınav Tarihi: 17.5.2015 17:35:00

Sınav Süresi (dk): 60

[Cosine Hesapla](#)

[Lsa Hesapla](#)

[Ilsa Hesapla](#)

Sınavda Bulunan Soru Sayısı: 10

Sınava Giren Toplam Öğrenci Sayısı: 41

COSINE SONUÇLARI

Sıra No	Öğrenci No	Ad Soyad	S1P	S2P	S3P	S4P	S5P	S6P	S7P	S8P	S9P	S10P	Toplam Puan
1	1325010013	ogr1	0	2,11	2,38	2,14	2,45	0,83	1,29	0,68	0,4	2,55	14,83
2	1325010014	ogr2	4,26	4,08	2,98	4,04	1,41	2,24	0	1,41	8,73	4,02	33,17
3	1325010015	ogr3	4,05	4,08	2,2	2,83	3,77	3,65	0	0,91	1,33	0	22,82
4	1325010016	ogr4	5,33	4,08	4,27	2,77	3,16	3,16	0	1,49	4,95	3,32	32,53
5	1325010017	ogr5	3,69	9,85	1,65	1,05	7,67	9,49	0	2,2	1,94	3,79	41,33
6	1325010018	ogr6	3,2	10	4,52	5,39	8,01	3,16	5,48	0,91	9,81	3,16	53,64
7	1325010019	ogr7	3,02	10	1,65	3,2	3,3	7,75	5,77	0	0	2,86	37,55
8	1325010020	ogr8	3,48	9,8	1,65	0	2,92	2,24	0	0,46	1,15	2,12	23,82
9	1325010022	ogr9	0	4,08	2,05	1,81	5,56	4,47	0	3,27	4,9	5,67	31,81
10	1325010023	ogr10	3,69	10	4,57	5,88	4,3	5,16	2,78	1	0,76	2,31	40,45
11	1325010024	ogr11	5,03	4,08	5,38	4,92	7,67	8,94	0	0	10	5,25	51,27
12	1325010025	ogr12	4,92	4,08	3,97	2,66	6,77	9,13	0	0,55	9,17	6,26	47,51

Şekil 4.3. Eğitimci için otomatik puanlama sonuçları ekran görüntüsü

TSPS yazılımında öğrenci işleri kullanıcıları, eğitimcilere ders atama işlemini ders işlemlerinden yapabilmektedir. Eğitimcilerin üzerindeki dersleri ve derslere kayıtlı öğrencileri görüntüleme işlemleri de öğrenci işleri kullanıcıları tarafından yapılmaktadır.

Öğrenciler derse kayıt olma, sınava girme ve sınavdaki kendi sonucunu görme işlemlerini yapmaktadır. Öğrenciler sınav sonuçlarını sorulara göre puan dağılımıyla beraber ve soruların cevap anahtarlarını da öğrenebilmektedir. Şekil 4.4'de öğrenci için otomatik sınav puanlama sistemi sonucu ekran görüntüsü sunulmuştur.

Sınav No:	15				
Sınav Adı:	yeni deneme				
Sınav Tarihi:	17.5.2015 17:35:00				
Sınav Süresi (dk):	60				
Sınavda Bulunan Soru Sayısı: 10					
Sınavdan Alınan Toplam Puan: 14,83					
● COSINE ● LSA ● ILSA					
Soru No	Soru	Doğru Cevap	Verilen Cevap	Soru Puan	Alınan Puan
1	OSI referans modelinin birince katmanındaki cihazları yazınız	network interface card yineleyici hub multi access unit kablo alıcı ve verici	modem switch	10	0
2	Ağ topoloji türlerini yazınız	doğrusal topoloji halka topoloji yıldız topoloji ağaç topoloji karmaşık topoloji	ağaç star bus	10	2,11
3	OSİ referans modelindeki veri iletim katmanının özelliklerini yazınız	ağ katmanından aldığı veri paketlerine hata kontrol bitlerini ekleyerek çerçeve frame halinde fiziksel katmana iletme işinden sorumludur iletilen çerçevenin doğru mu yoksa yanlış mı iletildiğini kontrol eder eğer çerçeve hatalı iletilmişse çerçevenin yeniden gönderilmesini sağlar ayrıca ağ üzerindeki diğer bilgisayarları tanımlama kablolu o anda kimin tarafından kullanıldığını tespitini yapar örn ethernet frame relay isdn switch ve bridge	verilerin iletilmesi için birden fazla cihazın bağlı olması lazım eğer bu durum sağlanıyorsa osi iletim katmanında veriler derlenir ve nereye hangi yolla nasıl gideceği karara bağlanır ve veri yola çıkar br	10	2,38
4	Doğrusal Topolojinin dezavantajlarını yazınız	omurga kabloda bir bozulma veya kesilme olursa tüm ağ bağlantısı kesilir kablunun sonunda sonlandırıcı terminator olmalıdır ağda sorun olduğunda sorunun Öner'den kaynaklandığını bulmak zaman alıcı olabilir tek başına tüm bir binanın ağ çözümü için genellikle kullanılmamaktadır çarpışma	bilgi tüm bilgisayarlara uğrar eğer tek bir bilgisayarda hata olursa tüm ağ çalışmaz gönderilen bilgiyi sadece gönderilmiş bilgisayar tarafından görülür	10	2,14
5	Topoloji nedir?	bir ağdaki bilgisayarların nasıl yerleşeceğini nasıl bağlanacağını veri iletiminin nasıl olacağını belirleyen genel yapıdır fiziksel topoloji ağın	topluluk diye biliriz yani bilgisayarların birbirlerine bağlı kalmasını ve iletişim içinde	10	2,45

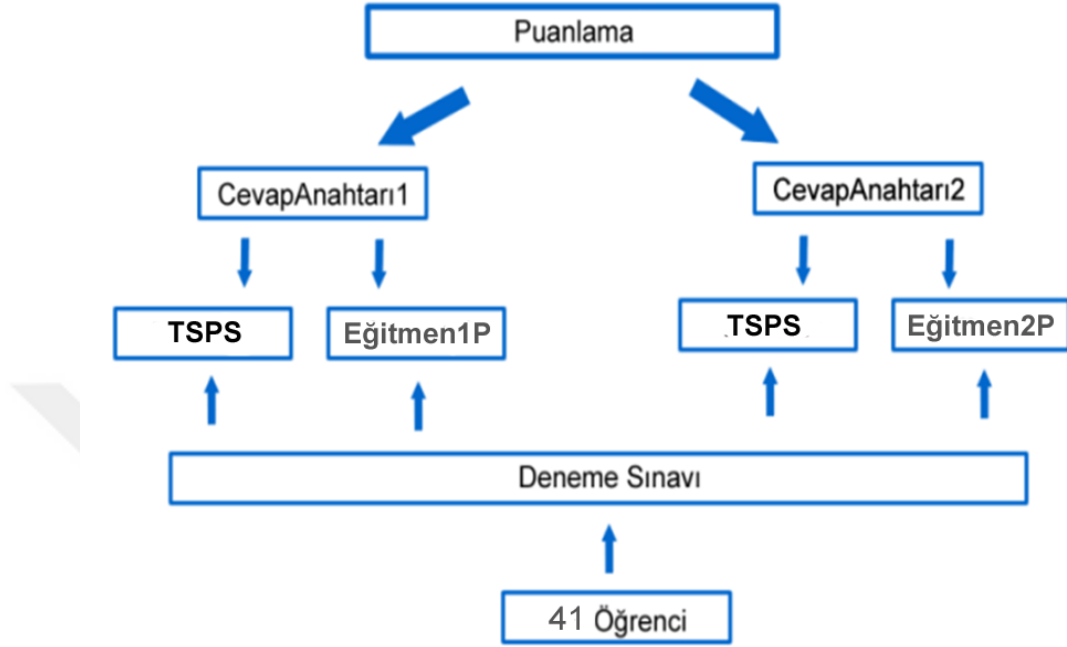
Şekil 4.4. Öğrenci için otomatik sınav puanlama sistemi sonucu ekran görüntüsü

4.2. TSPS Yazılımının Yeterlilik Testi

Literatür taraması yapıldığında, otomatik puanlama hakkında birçok çalışmaya rastlanmaktadır. PEG (Page, 1994), IEA (Landauer vd., 2000) ve E-rater (Burstein vd., 1998) otomatik sınav puanlama sistemlerinden birkaçıdır. Yapılan çalışmalardan hiçbiri, Türkçe dilindeki gerçekleştirilen sınavlar için yapılmamıştır. Bu açıdan, TSPS yazılımı Türkçe için geliştirilen ilk otomatik sınav değerlendirme yazılımıdır.

TSPS yazılımının öğrenci cevaplarına verdiği puanların doğruluğunu test etmek için, Süleyman Demirel Üniversitesi Uzaktan Eğitim Bilgisayar Mühendisliği öğrencileriyle örnek olay çalışması yapılmıştır. Bilgisayar ağları dersi ile alakalı

açık uçlu on soru hazırlanmış ve kırk bir öğrenci bu sınava katılmıştır. Sınav puanlamasının geçerliği ve güvenilirliği için hazırlanan on soru için, iki eğitimci farklı cevap anahtarları oluşturmuştur.



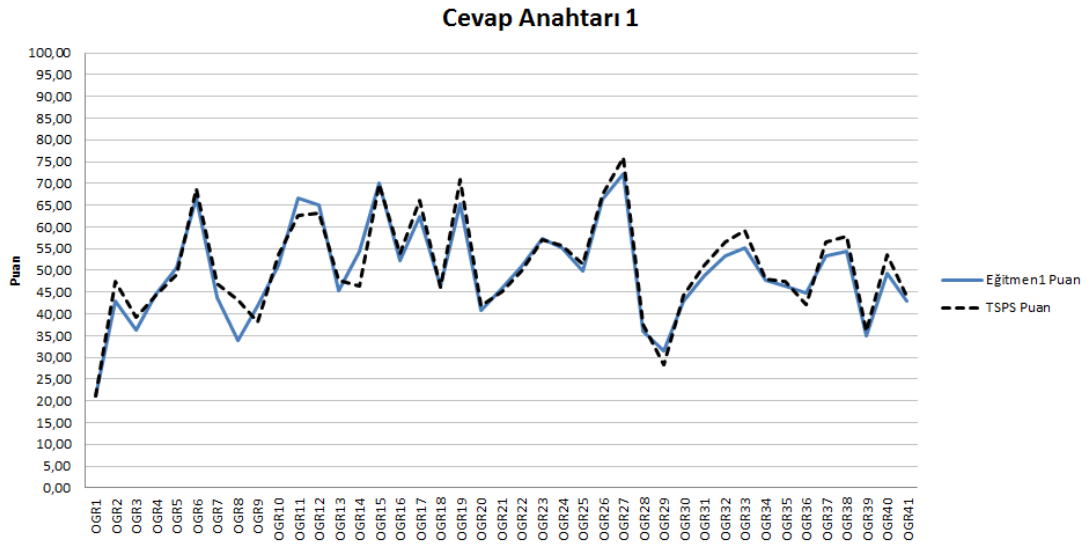
Şekil 4.5. Puanlamalar

10 soruluk deneme sınavını iki eğitimci beraber hazırlamıştır. Eğitimci sorular için hazırladıkları cevap anahtarlarını sisteme birbirinden bağımsız olarak girmişlerdir. Öğrenciler sistemde sınav olduktan sonra, öğrencilerin sınavları TSPS tarafından otomatik olarak değerlendirilmiştir. Birinci eğitimcinin sınav cevaplarını okumasıyla elde edilen puanlar Eğitimci1 ve ikinci eğitimcinin sınav cevaplarını okumasıyla elde edilen puanlar Eğitimci2 puanı olarak kaydedilmiştir. TSPS yazılımının birinci ve ikinci cevap anahtarına göre hesaplama sonucu elde edilen puanlar TSPS olarak kaydedilmiştir (Şekil 4.5).

TSPS yazılımı, kosinüs benzerliği ve TGAA metodundan oluşan hibrid benzerlik ölçütü ile sınav sonuçlarını hesaplamaktadır. Eğitimcinin sınav cevap anahtarındaki kelime sayısına bakılarak benzerlik hesabına sistem otomatik karar vermektedir. Cevap anahtarı 10 kelimedenden az olma durumunda GAA hesaplaması için yeterli boyut oluşmayacağından dolayı (Landauer ve Dumais,

1997; Kontostathis, 2007), sistem otomatik olarak kosinüs benzerliğini seçmektedir. Kelime sayısı 10'dan fazla ise GAA tabanlı TGAA metodu kullanılmaktadır. Bu hibrid yapıya, sistemin farklı iki cevap anahtarına göre kosinüs, GAA ve TGAA hesaplama sonuçları dikkate alınarak karar verilmiştir. Sistemin birinci cevap anahtarına göre kosinüs benzerlik, GAA ve TGAA hesabına göre verdiği puanlar EK M'de verilmiştir. Sistemin ikinci cevap anahtarına göre kosinüs benzerlik, GAA ve TGAA hesabına göre verdiği puanlar EK N'de verilmiştir.

EK M ve EK N'deki puanların dağılımının incelenmesinin yanı sıra, soruların cevap anahtarı kelime uzunluklarına göre kosinüs benzerliği veya TGAA hesaplama yönteminden hangisinin kullanılacağı tespit edilmiştir. Eğitimci1 ve Eğitimci2'nin sorulara hazırladıkları cevap anahtarları EK O ve EK Ö'de verilmiştir. TSPS hibrid yapısının oluşturulması ve sistemin benzerlik tercihinin belirlenmesi işlemlerinden sonra TSPS yazılımının başarısı test edilmiştir (Şekil 4.6).



Şekil 4.6. Eğitimci1 ve TSPS puanları karşılaştırması

TSPS yazılımının başarısını analiz etmek ve değerlendirmek için, eğitimciler kendi cevap anahtarlarına göre öğrencilerin sınav cevaplarını değerlendirip puanlamışlardır. Şekil 4.6'da Eğitimci1 ve TSPS puanları grafiksel olarak sunulmuştur. Ayrıca Çizelge 4.1'de Eğitimci1 ve TSPS puanlarının her soru için

değerlendirici ortalaması ve standart sapması değerleri sunulmuştur. Her soru on puan üzerinden değerlendirilmiştir.

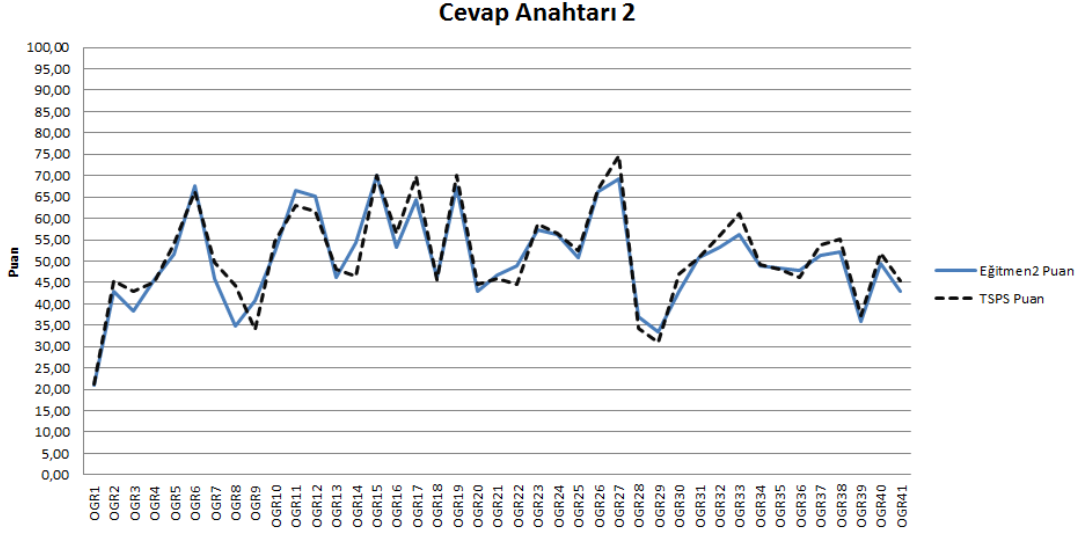
Çizelge 4.1. Eğitimci ve TSPS puanları

	Değerlendirme	Katılımcı		Standart Sapma	Standart
		Sayısı	Ortalama		Hata Ortalaması
soru1	Eğitmen1P	41	3,9317	1,93822	,30270
	TSPS	41	4,2973	1,90044	,29680
soru2	Eğitmen1P	41	6,9268	2,56311	,40029
	TSPS	41	6,5024	3,03519	,47402
soru3	Eğitmen1P	41	6,2683	3,04979	,47630
	TSPS	41	6,8932	3,36127	,52494
soru4	Eğitmen1P	41	6,0976	3,33770	,52126
	TSPS	41	5,8483	3,76330	,58773
soru5	Eğitmen1P	41	6,1463	2,46537	,38503
	TSPS	41	6,0702	2,55523	,39906
soru6	Eğitmen1P	41	5,9512	2,52934	,39502
	TSPS	41	6,0727	2,46405	,38482
soru7	Eğitmen1P	41	1,2683	2,03745	,31820
	TSPS	41	1,3112	2,08959	,32634
soru8	Eğitmen1P	41	1,7561	2,46759	,38537
	TSPS	41	2,0080	2,45713	,38374
soru9	Eğitmen1P	41	3,2927	3,05159	,47658
	TSPS	41	3,4112	3,15067	,49205
soru10	Eğitmen1P	41	8,1220	2,39995	,37481
	TSPS	41	8,4644	2,22029	,34675

Şekil 4.6'da sunulan Eğitimci ve TSPS puanlarının grafik eğrilerinin benzer olduğu, grafikler arasında çok küçük farklılıklar olduğu tespit edilmiştir. Birinci eğitmenin sınav değerlendirmesi ile TSPS yazılımının sınav değerlendirmesinin yakın olduğu bulunmuştur. Ayrıca Çizelge 4.1'de Eğitimci ve TSPS puanları sorulara göre incelendiğinde, farklı değerlendiricilerin ilgili sorular için puan ortalamalarının yakın olduğu tespit edilmiştir. Bu durum TSPS yazılımının, birinci eğitmenin verdiği puanlara yakın puanlama yaptığını göstermektedir.

İkinci eğitmen de aynı sorular için cevap anahtarı hazırlamış ve TSPS yazılımına bu soruların cevaplarını kaydetmiştir. İkinci eğitmenin sınav değerlendirmesi Eğitimci2 puanları ile TSPS puanları karşılaştırması Şekil 4.7'de grafiksel olarak

sunulmuştur. Bu aşamada cevap anahtarı ikiye göre değerlendirme yapılmıştır. İkinci eğitmenin sınav değerlendirmesi ile TSPS yazılımının sınav değerlendirmesinin yakın olduğu tespit edilmiştir.



Şekil 4.7. Eğitmen2 ve TSPS puanları karşılaştırması

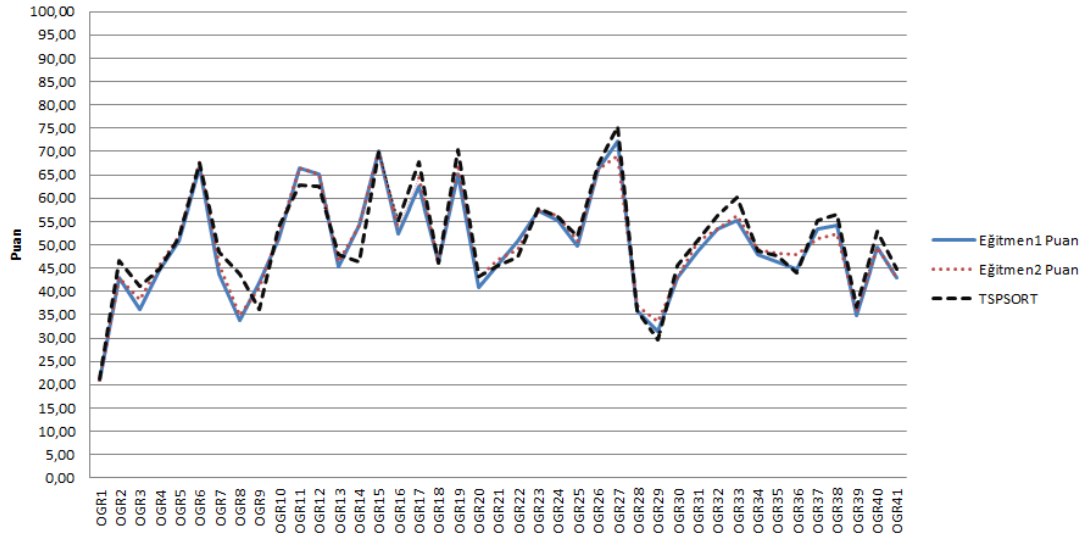
Çizelge 4.2’de Eğitmen2 ve TSPS puanlarının her soru için değerlendirici ortalaması ve standart sapması değerleri sunulmuştur. Ayrıca Çizelge 4.2’de Eğitmen2 ve TSPS puanları sorulara göre incelendiğinde, farklı değerlendiricilerin ilgili sorular için puan ortalamalarının yakın olduğu tespit edilmiştir. Bu durum TSPS yazılımının, ikinci cevap anahtarını kullanarak, ikinci eğitmenin verdiği puanlara yakın puanlama yaptığını göstermektedir.

Çizelge 4.2. Eğitim2 ve TSPS puanları

	Değerlendirme	Katılımcı Sayısı	Ortalama	Standart Sapma	Standart
					Hata Ortalaması
soru1	Eğitmen2P	41	3,9317	1,93822	,30270
	TSPS	41	4,2973	1,90044	,29680
soru2	Eğitmen2P	41	6,9268	2,56311	,40029
	TSPS	41	6,5024	3,03519	,47402
soru3	Eğitmen2P	41	6,2683	3,08240	,48139
	TSPS	41	6,7707	3,43006	,53569
soru4	Eğitmen2P	41	6,0976	3,33770	,52126
	TSPS	41	6,1332	3,69839	,57759
soru5	Eğitmen2P	41	6,9512	2,67349	,41753
	TSPS	41	6,9422	2,63305	,41121
soru6	Eğitmen2P	41	5,9512	2,52934	,39502
	TSPS	41	6,0727	2,46405	,38482
soru7	Eğitmen2P	41	1,2683	2,03745	,31820
	TSPS	41	1,3112	2,08959	,32634
soru8	Eğitmen2P	41	1,6341	2,09471	,32714
	TSPS	41	1,7154	2,12088	,33123
soru9	Eğitmen2P	41	3,1707	2,80091	,43743
	TSPS	41	2,9863	2,36615	,36953
soru10	Eğitmen2P	41	8,1220	2,37902	,37154
	TSPS	41	8,4968	2,38123	,37189

Eğitmenlerin puanları ile TSPS yazılımı hesaplamaları arasındaki en iyi uyumu tespit etmek için, aritmetik ortalama puanı hesaplanmış olup, bu yeni puan TSPSORT olarak kaydedilmiştir. Aritmetik ortalama hesabı birinci ve ikinci cevap anahtarına göre hesaplanan TSPS puanları kullanılarak yapılmıştır. TSPSORT ile Eğitim1 ve Eğitim2 puanları ile karşılaştırması Şekil 4.8’de sunulmuştur. Şekil 4.8’deki grafik incelendiğinde TSPSORT puanının Eğitim1 ve Eğitim2 puanları ile uyumlu olduğu, öğrencilerin puanlarını temsil edebileceği tespit edilmiştir. Böylece TSPS yazılımının öğrenciye vereceği sınav nihai sonuç puanı TSPSORT olarak belirlenmiştir. Çizelge 4.3’de Eğitim1 ve TSPSORT, Çizelge 4.4’de ise Eğitim2 ve TSPSORT puanlarının her soru için değerlendirici ortalaması ve standart sapması değerleri sunulmuştur.

Aritmetik Ortalama Puanı



Şekil 4.8. Eğitim1, Eğitim2 ve TSPSORT puanları karşılaştırması

Çizelge 4.3. Eğitim1 ve TSPSORT puanları

Soru	Değerlendirme	Katılımcı		Standart Sapma	Standart Hata Ortalaması
		Sayısı	Ortalama		
soru1	Eğitmen1P	41	3,9317	1,93822	,30270
	TSPSORT	41	4,2973	1,90044	,29680
soru2	Eğitmen1P	41	6,9268	2,56311	,40029
	TSPSORT	41	6,5024	3,03519	,47402
soru3	Eğitmen1P	41	6,2683	3,04979	,47630
	TSPSORT	41	6,8320	3,39086	,52956
soru4	Eğitmen1P	41	6,0976	3,33770	,52126
	TSPSORT	41	5,9907	3,70533	,57868
soru5	Eğitmen1P	41	6,1463	2,46537	,38503
	TSPSORT	41	6,5062	2,54544	,39753
soru6	Eğitmen1P	41	5,9512	2,52934	,39502
	TSPSORT	41	6,0727	2,46405	,38482
soru7	Eğitmen1P	41	1,2683	2,03745	,31820
	TSPSORT	41	1,3112	2,08959	,32634
soru8	Eğitmen1P	41	1,7561	2,46759	,38537
	TSPSORT	41	1,8617	2,27473	,35525
soru9	Eğitmen1P	41	3,2927	3,05159	,47658
	TSPSORT	41	3,1988	2,72786	,42602
soru10	Eğitmen1P	41	8,1220	2,39995	,37481
	TSPSORT	41	8,4806	2,28988	,35762

Çizelge 4.4. Eğitim2 ve TSPSORT puanları

	Değerlendirme	Katılımcı		Standart Sapma	Standart Hata Ortalaması
		Sayısı	Ortalama		
soru1	Eğitmen2P	41	3,9317	1,93822	,30270
	TSPSORT	41	4,2973	1,90044	,29680
soru2	Eğitmen2P	41	6,9268	2,56311	,40029
	TSPSORT	41	6,5024	3,03519	,47402
soru3	Eğitmen2P	41	6,2683	3,08240	,48139
	TSPSORT	41	6,8320	3,39086	,52956
soru4	Eğitmen2P	41	6,0976	3,33770	,52126
	TSPSORT	41	5,9907	3,70533	,57868
soru5	Eğitmen2P	41	6,9512	2,67349	,41753
	TSPSORT	41	6,5062	2,54544	,39753
soru6	Eğitmen2P	41	5,9512	2,52934	,39502
	TSPSORT	41	6,0727	2,46405	,38482
soru7	Eğitmen2P	41	1,2683	2,03745	,31820
	TSPSORT	41	1,3112	2,08959	,32634
soru8	Eğitmen2P	41	1,6341	2,09471	,32714
	TSPSORT	41	1,8617	2,27473	,35525
soru9	Eğitmen2P	41	3,1707	2,80091	,43743
	TSPSORT	41	3,1988	2,72786	,42602
soru10	Eğitmen2P	41	8,1220	2,37902	,37154
	TSPSORT	41	8,4806	2,28988	,35762

Değerlendiriciler arası uyumu hesaplamak için, Eğitim1, Eğitim2 ve TSPSORT puanları arasında Kendall W hesaplaması yapılmıştır. Kendall W (Kendall ve Babington-Smith, 1939) hesabında “0” sonucu değerlendiriciler arasında hiç uyum olmadığını, “1” sonucu ise değerlendiricilerin aynı cevaba aynı puanı verdikleri yanı aralarındaki uyumun maksimum olduğunu belirtmektedir. Hesaplama m tane hesaplayıcı, k kadar konuyu 1’den k ’ya kadar değerlendirme yapıldığında, derecelendirme değerlendiricisi j konu i ’ye vererek r_{ij} olarak ifade edilir. Her konu için ise, $R_i = \sum_{j=1}^m r_{ij}$ ve \bar{R} hesaplanır, \bar{R} R_i değerlerinin ortalamasıdır. Kendall W hesabı formül 4.1a ve 4.1b’de verilmiştir.

$$R = \sum_{i=1}^k (R_i - \bar{R})^2 \quad (4.1a)$$

$$W = \frac{12R}{m^2(k^3 - k)} \quad (4.1b)$$

Eđitmen1, Eđitmen2 ve TSPSORT puanları arasında Kendall W sonucu 0.9254'dir ve bu deęer deęerlendiriciler arasında mükemmel derecede uyum olduęunu belirtmektedir. Ayrıca Kendall W hesabı TSPS1, Eđitmen1 ve Eđitmen2 arasında hesaplanmış sonuç 0.9204 bulunmuştur ve TSPS2, Eđitmen1 ve Eđitmen2 arasında Kendall W sonucu 0.9174 olarak bulunmuştur. Sonrasında Pearson's t test puanlar arasında uygulanmıştır ve TSPSORT ile Eđitmen1 veya Eđitmen2 arasında anlamlı bir farklılık ($p>0.05$) bulunmamıştır (Çizelge 4.5 ve Çizelge 4.6) (EK P, EK R, EK S, EK Ş).

Çizelge 4.5. Eđitmen1 ve TSPSORT puanları t testi sonuçları

	Deęerlendirme	Ortalama	t	p
soru1	Eđitmen1P	3,9317	-0,862	0,391
	TSPSORT	4,2973		
soru2	Eđitmen1P	6,9268	0,684	0,496
	TSPSORT	6,5024		
soru3	Eđitmen1P	6,2683	-0,791	0,431
	TSPSORT	6,8320		
soru4	Eđitmen1P	6,0976	0,137	0,891
	TSPSORT	5,9907		
soru5	Eđitmen1P	6,1463	-0,65	0,517
	TSPSORT	6,5062		
soru6	Eđitmen1P	5,9512	-0,22	0,826
	TSPSORT	6,0727		
soru7	Eđitmen1P	1,2683	-0,094	0,925
	TSPSORT	1,3112		
soru8	Eđitmen1P	1,7561	-0,201	0,841
	TSPSORT	1,8617		
soru9	Eđitmen1P	3,2927	0,147	0,884
	TSPSORT	3,1988		
soru10	Eđitmen1P	8,1220	-0,692	0,491
	TSPSORT	8,4806		

Çizelge 4.6. Eğitim2 ve TSPSORT puanları t testi sonuçları

	Değerlendirme	Ortalama	t	p
soru1	Eğitmen2P	3,9317	-0,862	0,391
	TSPSORT	4,2973		
soru2	Eğitmen2P	6,9268	0,684	0,496
	TSPSORT	6,5024		
soru3	Eğitmen2P	6,2683	-0,788	0,433
	TSPSORT	6,8320		
soru4	Eğitmen2P	6,0976	0,137	0,891
	TSPSORT	5,9907		
soru5	Eğitmen2P	6,9512	0,772	0,442
	TSPSORT	6,5062		
soru6	Eğitmen2P	5,9512	-0,220	0,826
	TSPSORT	6,0727		
soru7	Eğitmen2P	1,2683	-0,094	0,925
	TSPSORT	1,3112		
soru8	Eğitmen2P	1,6341	-0,471	0,639
	TSPSORT	1,8617		
soru9	Eğitmen2P	3,1707	-0,046	0,963
	TSPSORT	3,1988		
soru10	Eğitmen2P	8,1220	-0,695	0,489
	TSPSORT	8,4806		

5. SONUÇ VE ÖNERİLER

Literatür incelendiğinde, her otomatik değerlendirme sistemi kendi eğitim diline uygun şekilde geliştirilmiş olup bunlar içerisinde en çok kullanılan dil İngilizce'dir (Mohler vd., 2011). Sunulan tezde, web tabanlı Türkçe otomatik sınav puanlama sistemi yazılımı geliştirilmiş ve yazılım gerçek bir deneme sınavında kullanılmıştır. Bu çalışmanın özgünlüğü, Türkçe için ilk otomatik sınav puanlama sistemi olmasıdır.

TSPS yazılımı, öğretmen cevap anahtarındaki kelime sayısını dikkate alarak hibrid yapılı bir benzerlik ölçütü ile sınav sonuçlarını yüksek başarıyla hesaplamaktadır. Bu hibrid yapıda kosinüs benzerliği ve TGAA metodu kullanılmaktadır. TGAA metodu bu çalışmada geliştirilmiş olup, GAA hesaplamasını kullanmaktadır. TGAA metodunda GAA metodundan farklı olarak eş anlam, alt kavram, kısaltma ve rakam durumları vektör oluşturulmadan önce tespit edilmektedir. Bu durum metinler arası anlamsal ilişkilerin belirlenmesi açısından, TGAA'nın GAA'ya göre daha başarılı olmasını sağlamaktadır. Ayrıca anlambilimsel analiz yapan TGAA yöntemi ile gizli anlambilimsel çekirdek yöntemi karşılaştırılması yapılmıştır. Karşılaştırma sonuçlarına göre, TGAA yöntemi kullanılarak elde edilen puanların, gizli anlambilimsel çekirdek yöntemi puanlarına göre, öğretmen puanlarına daha yakın olduğu tespit edilmiştir.

TSPS yazılımının başarısının değerlendirilmesi için örnek olay çalışması yapılmıştır. Örnek olay çalışmasında yazılımın başarısı, öğretmenlerin puanları ve TSPS yazılımı puanı ile karşılaştırılarak tespit edilmiştir. Puanlama doğruluğunu arttırmak için, aynı sorular için iki öğretmen farklı cevap anahtarları hazırlamışlardır. Puanlar arası karşılaştırmalar grafiklerle sunulmuş, puanlamaların yakınlığı ve TSPS yazılımının başarısı betimlenmiştir. Ayrıca TSPS yazılımının farklı cevap anahtarları için hesapladığı sonuçların aritmetik ortalaması alınarak, değerlendirme sonuç puanı belirlenmiştir.

Eđitmenler, TSPS yazılımının puanlandırmasında en iyi uyumu gerekleřtirmek iin iki veya daha fazla cevap anahtarları ile alıřabilmektedir. rnek olay alıřmasında gerekleřtirilen sınav puanlama sonuları, Kendall *W* hesabıyla TSPS yazılımı bařarısı %92 bulunmuřtur. Sonu olarak, dođal dil iřleme algoritmalarını kullanan TSPS yazılımı, aık ulu sorular ieren bilgisayar ađları sınavları iin kullanılmaya uygundur.

Eđitmenler TSPS yazılımını, evrimii sınav yapma, evrimii dev kontrol etme ve deneme sınavları iin kullanabilmektedir. TSPS yazılımı kısa cevaplı soruların depolanmasını ve yapılan sınavların otomatik olarak deđerlendirilmesini sađlamaktadır. Eđer Trke WordNet zenginleřtirilirse, TSPS yazılımı diđer dersler ve diđer eđitim kurumları tarafından da kullanılabilir. nk TSPS yazılımının sınav puanlaması yapmak iin, puanlama yapılacak dersin teknik terimlerinin Trke WordNet'de bulunması gerekmektedir. Bu durum herhangi bir sınavı okuyacak kiřinin o sınav dersinin teknik terimlerine hkim olması gerekliliđi ile aynıdır.

Gelecek alıřmalarda, web servisleri kullanılarak, TSPS yazılımının diđer eđitim kurumlarına eđitsel kullanım amalı olarak aılması planlanmaktadır. Bunun yanı sıra, ontolojik yaklařım (Zhang vd., 2008), wikipedia (Vikipedi, 2015) tabanlı aık anlambilimsel analiz (Gabrilovich ve Markovitch, 2009; Rahutomo ve Aritsugi, 2014) veya gizli anlambilimsel analiz trevleri (Haralambous ve Klyuev, 2013) gibi diđer dođal dil iřleme metotlarının kullanımının denenmesi ve gncelleřtirilmesi dřnlmektedir. Geliřtirilecek yeni srmlerin sunumu iin proje web sayfası (TSPS, 2015) oluřturulmuřtur.

KAYNAKLAR

- Adalı, Ş., 2009. An Integrated Architecture for Information Extraction from Documents in Turkish. Istanbul Technical University, Institute of Science and Technology, Doctorate Thesis, 109, İstanbul.
- Agun, H.V., 2008. Doğal Dil İşlemede Çizgisel ve Olasılık Tabanlı bir Otomatik Öğrenme Uygulaması. Trakya Üniversitesi, Fen Bilimleri Enstitüsü, Master Tezi, 59, Edirne.
- Akın, M. D., Akın, A. A., 2007. 'An Open Source Natural Language Processing Library for Turkic Languages: Zemberek'. Electrical Engineering, 2007, 431, pp 38
- Akman, Ç.K., 2004. Computer-Aided Transcription Tool. Boğaziçi University, Graduate School of Natural and Applied Sciences, Master Thesis, 75, İstanbul.
- Aksoy, E., 2008. HPSG Teorisinin ve Semantik Frame Teorisinin Bir Uygulaması Olarak, Sahne Betimleyen Doğal Dil Cümlelerinden Görsel Yapılar İnşa Edilmesi. Trakya Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 91, Edirne.
- Aktaş, Ö., 2006. Türkçe için Verimli bir Cümle Sonu Belirleme Yöntemi. Akademik Bilişim 2006 - Bilgi Teknolojileri Kongresi IV, Pamukkale University, Denizli, Türkiye.
- Albayrak, N.B., 2011. Doğal Dil İşleme Teknikleri Kullanarak Görüş ve Duygu Analizi. Fatih University, Institute of Sciences and Engineering, Master Thesis, 69, İstanbul.
- Alberto, B. , Paolo, R., Eneko A., Gorika L., 2010. Plagiarism Detection across Distant Language Pairs, In Proceedings of the 23rd International Conference on Computational Linguistics, pages 37-45.
- Alkım, E., 2006. Türkçe için Doğal Dil İşlemede Biçimbirimsel Çözümleme ve Sözlük Tasarımı için Yeni Bir Yöntem. Dokuz Eylül University. Graduate School of Natural and Applied Sciences, Master Thesis, 81, İzmir.
- Attali, Y., Burstein, J. C., 2006. Automated essay scoring with e-rater v.2. Journal of Technology Learning, and Assessment, 4 (3).
- Bahadır, Ö., 2007. Aritmetik Problemlerin Çözümlemesi ve Simgelemesi. Istanbul Technical University, Institute of Science and Technology, Master Thesis, 66, İstanbul.
- Bay, Ö.F., Tüzün, H., 2002. Yüksek Öğretim Kurumlarında Ders İçeriğinin Web Tabanlı Olarak Aktarılması-I, Journal of Polytechnic, 5 (1), 13-22.

- Beldarrain, Y., 2006. Distance education trends: Integrating new technologies to foster student interaction and collaboration. *Distance education*, 27(2), 139-153.
- BETSY, 2014. <http://www.edres.org/betsy/history.htm> Son Erişim Tarihi: 10.09.2014
- Bilgin, O., Çetinoğlu, Ö., Oflazer, K., 2004. Building a wordnet for Turkish. *Romanian Journal of Information Science and Technology*, 7(1-2), 163-172.
- Birant, Ç.C., 2008. Root-Suffix Separation of Turkish Words. Dokuz Eylül University, Graduate School of Natural and Applied Sciences, Master Thesis", 56, İzmir.
- Bochkanov, S., Bystritsky, V., 2011. ALGLIB-a cross-platform numerical analysis and data processing library. ALGLIB Project, Novgorod, Russia, 2011
- Brown, A.L., 1992. Design experiments: Theoretical and methodological challenges in creating complex interventions in classroom settings. *Journal of the Learning Sciences*, 2(2), 141-178.
- Burstein, J. C., 2003. The e-rater® scoring engine: Automated essay scoring with natural language processing. In: M. D. Shermis, J. C. Burstein (Eds.), *Automated essay scoring: A cross-disciplinary perspective* (pp. 113-121). Hillsdale, NJ: Erlbaum.
- Burstein, J., Kaplan, R., Wolff, S., Lu, C., 1996. Using Lexical Semantic Techniques to Classify Free-Responses. *Proceedings from the SIGLEX96 Workshop*, ACL, University of California, Santa Cruz.
- Burstein, J., Kukich, K., Wolff, S., Lu, C., Chodorow, M., 1998. Computer analysis of essays. Paper presented at the annual meeting of the National Council of Measurement in Education. San Diego, CA. <http://www.ets.org/research/dload/ncmefinal.pdf> Son Erişim Tarihi: 10.09.2014
- Burstein, J., Kukich, K., Wolff, S., Lu, C., Chodorow, M., 1998. Enriching Automated Essay Scoring Using Discourse Marking. *Proceedings of the Workshop on Discourse Relations and Discourse Markers*, Annual Meeting of the Association of Computational Linguistics, August, Montreal, Canada.
- Burstein, J., Chodorow, M., 1999. Automated Essay Scoring for nonnative English speakers. *Proceedings of the ACL99 Workshop on Computer Mediated Language Assessment and Evaluation of Natural Language Processing*, College Park, MD.

- Burstein, J., Marcu, D., 2000. Benefits of modularity in an automated essay scoring system. In Proceedings of the COLING-2000 Workshop on Using Toolsets and Architectures To Build NLP Systems (pp. 44-50). Association for Computational Linguistics.
- Burstein, J., Leacock, C., Swartz, R., 2001. Automated Evaluation of Essay and Short Answers. In: M. Danson (ed.) Proceedings of the Sixth International Computer Assisted Assessment Conference, Loughborough University, Loughborough, UK.
- Christie, J. R., 1999. Automated Essay Marking – For Both Style and Content. In: Danson, M. (ed.) Proceedings of the Third International Computer Assisted Assessment Conference, Loughborough University, Leicestershire, UK.
- Chung, K. W. K., O’Neil, H. F., 1997. Methodological approaches to online scoring of essays (ERIC reproduction service no ED 418 101).
- Cilibrasi, R.L., Vitanyi, P.M.B., 2007. The Google Similarity Distance, IEEE Trans. Knowledge and Data Engineering, 19:3, 370-383.
- Cristianini, N., Shawe-Taylor, J., Lodhi, H., 2002. Latent semantic kernels. Journal of Intelligent Information Systems, 2002, 18(2-3), 127-152
- Collins, A., 1992. Towards a design science of education. E. Scanlon ve T. O’Shea (Eds.), New directions in educational technology, p.15-22, Berlin: Springer.
- Çakıroğlu, Ü., 2006. Türkçe Aritmetik Problemlerin Bilgisayarla Çözümü. Eleco, Elektrik-Elektronik-Bilgisayar Mühendisliği Sempozyumu ve Fuarı, Bursa.
- Çölkesen, R., Örencik B., 2003. Bilgisayar Haberleşmesi ve Ağ Teknolojileri, 1448s., İstanbul, 2003.
- Damashek, M., 1995. Gauging Similarity with N-grams: Language-independent Categorization of Text. Science, 267(5199), 843-848.
- Delibaş, A., 2008. Doğal Dil İşleme ile Türkçe Yazım Hatalarının Denetlenmesi. İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 78, İstanbul.
- Dice, L., 1945. Measures of the amount of ecologic association between species. Ecology, 26(3).
- Elliot, S., 2003. IntelliMetric: from here to validity. In Mark D. Shermis and Jill C. Burstein (Eds.). Automated essay scoring: a cross disciplinary approach. Mahwah, NJ: Lawrence Erlbaum Associates.

- Eker, Ö., 2009. Parser Evaluation Using Textual Entailments. Boğaziçi University, Master Thesis, 107, İstanbul.
- Er, N.P., 2009. Turkish Factoid Question Answering Using Answer Pattern Matching. Bilkent University, The Institute of Engineering and Sciences, Master Thesis, 155, Ankara.
- Eryiğit, G., 2006. Türkçe'nin Bağlılık Ayırıştırması. İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, 128, İstanbul.
- Eş, S., 2005. A Computational Analysis of a Language Structure in Natural Language Text Processing. Çankaya University, The Graduate School of Natural and Applied Sciences, Master Thesis, 52, Ankara
- Eugene F. K., 1987. Taxicab Geometry , Dover. ISBN 0-486-25202-7.
- Gabrilovich E., Markovitch, S., 2007. Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis, Proceedings of the 20th International Joint Conference on Artificial Intelligence, pages 6–12.
- Gabrilovich, E., Markovitch, S., 2009. Wikipedia-based semantic interpretation for natural language processing. Journal of Artificial Intelligence Research, 2009, 34 (2), pp 443–498
- Güngör, T., 1995. Computer Processing of Turkish: Morphological and Lexical Investigation. Boğaziçi University, Doktora Tezi, 185, İstanbul.
- Görmez, Z., 2009. Implementation of a Text-to-Speech System with Machine Learning Algorithms in Turkish. Fatih University, Institute of Sciences and Engineering, Master Thesis, 63, İstanbul.
- Hadımlı, K., 2011. Processing Turkish Radiology Reports. Middle East Technical University, The Graduate School of Natural and Applied Sciences, Master Thesis, 86, Ankara.
- Hallaç, Ü., 2007. Determination of Turkish Word Types. Dokuz Eylül University, Graduate School of Natural and Applied Sciences. Master Thesis, 51, İzmir.
- Haralambous, Y., Klyuev, V., 2013 Thematically reinforced explicit semantic analysis. International Journal of Computational Linguistics and Applications, 4(1):79–94, 2013.
- Hatzilygeroudis, I., Giannoulis, C., Koutsojannis, C., 2004. A Web-Based Education System for Predicate Logic, Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT'04).

- Hall , P. A. V., Dowling, G. R., 1980. Approximate string matching, *Comput. Surveys*, 12:381-402.
- Hebb, D., 1949. *The Organization of Behavior*. Wiley, p. 378, Newyork.
- Idea Works, 2013. <https://www.sagrader.com/>. Son erişim tarihi: 02.02.2014.
- Islam, A. and Inkpen, D., 2006. Second Order Cooccurrence PMI for Determining the Semantic Similarity of Words, in *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2006)*, Genoa, Italy, pp. 1033–1038.
- Islam, A. and Inkpen, D., 2008. Semantic text similarity using corpus-based word similarity and string similarity. *ACM Trans. Knowl. Discov. Data* 2, 2 (Jul. 2008), 1–25.
- Jaccard, P., 1901. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin de la Société Vaudoise des Sciences Naturelles* 37, 547-579.
- Jaro, M. A., 1989. Advances in record linkage methodology as applied to the 1985 census of Tampa Florida, *Journal of the American Statistical Society*, vol. 84, 406, pp 414-420.
- Jaro, M. A., 1995. Probabilistic linkage of large public health data file, *Statistics in Medicine* 14 (5-7), 491-8.
- İlkuçar, M., 2015. Detection of Similarity Rate of Compiler Independent Text Based Computer Programming Assignment and Homework Grading. *International Journal of Social Sciences and Education Research*, 1 (4), 1451-1459.
- İşman, A., 2008. *Uzaktan Eğitim*, Sayfa 255. PegemA Yayıncılık, Ankara.
- Kalender, M., 2010. *Automated Semantic Tagging of Text Documents*. Boğaziçi University, Master Thesis, 107, İstanbul.
- Karadeniz, Z.İ., 2007. *Türkçe için Biçimbirimsel Belirsizlik Giderici*. İstanbul Teknik Üniversitesi. Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 67, İstanbul.
- Kazkılınç, S., 2012. *Türkçe Metinlerin Etiketlenmesi*. İstanbul Teknik Üniversitesi. Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 63, İstanbul.
- Kendall, M. G., Babington-Smith B., 1939. The Problem of m Rankings. *The Annals of Mathematical Statistics*, 10 (3): 275- 287.

- Kışla, T., 2009. Türkçe için Tümlleşik bir Biçim Çözümleme ve Sözcük Türü Tespit Yöntemi. Ege Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, 185, İzmir.
- Kim, Y. B., Kim, Y. S., 2008. Latent semantic kernels for WordNet: Transforming a tree-like structure into a matrix. In *Advanced Language Processing and Web Information Technology*, 2008. ALPIT'08. International Conference on, 76-80, IEEE.
- Kontostathis, A., 2007. Essential dimensions of latent semantic indexing (lsi). In: *System Sciences*, 2007. HICSS 2007. 40th Annual Hawaii International Conference on. IEEE, 2007. p. 73-73.
- Köprü, S., 2008. Coupling Speech Recognition and Rule-Based Machine Translation. The Middle East Technical University, The Institute of Informatics, Doktora Thesis, 130, Ankara.
- Kukich, K., 2000. Beyond Automated Essay Scoring. *IEEE Intelligent Systems*, 15(5), 22-27.
- Kutlu, M., 2010. Noun Phrase Chunker for Turkish Using Dependency Parser. Bilkent University, The Institute of Engineering and Sciences, Master Thesis, 124, Ankara.
- Landauer, T.K., Dumais, S.T., 1997. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge, *Psychological Review*, 104.
- Landauer, T. K., Foltz, P. W., Laham, D., 1998. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3), 259-284.
- Landauer, T. K., Laham, D., Foltz, P. W., 2000. The Intelligent Essay Assessor. In M. A. Hearst (Ed.), *The debate on automated essay grading*. IEEE Intelligent systems, 27-31.
<http://que.infoscience.uiowa.edu/~light/research/mypapers/autoGradingIEEE.pdf>
- Landauer, T. K., Laham, D., Foltz, P. W., 2003. Automated scoring and annotation of essays with the Intelligent Essay Assessor. In: M. D. Shermis, J. C. Burstein (Eds.), *Automated essay scoring: A cross-disciplinary perspective* (pp. 87-112). Hillsdale, NJ: Erlbaum.
- Larkey, L. S., 1998. Automatic Essay Grading Using Text Categorization Techniques. *Proceedings of the Twenty First Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Melbourne, Australia, pp. 90-95.
- Leacock, C., Chodorow, M., 2003. C-rater: Automated scoring of short-answer questions. *Computers and the Humanities*, 37(4), 389-405.

- Lemaire, B., Dessus, P., 2001. A system to assess the semantic content of student essays. *Educational Computing Research*, 24(3), 305–306.
- Lund, K., Burgess, C., Atchley, R. A., 1995. Semantic and associative priming in a high-dimensional semantic space. *Cognitive Science Proceedings (LEA)*, 660-665.
- Lund, K., Burgess, C., 1996. Producing highdimensional semantic spaces from lexical cooccurrence. *Behavior Research Methods, Instruments & Computers*, 28(2),203-208.
- Mackay, D., 2013. The Dasher Project,
<http://www.inference.phy.cam.ac.uk/dasher/>. Son erişim tarihi: 16.07.2013.
- Martin, P., Benno, S., Maik, A., 2008. A Wikipediabased multilingual retrieval model. *Proceedings of the 30th European Conference on IR Research (ECIR)*, pp. 522-530.
- Mason, O., Grove-Stephenson, I., 2002. Automated Free Text Marking with Paperless School. In: Danson, M. (ed.) *Proceedings of the Sixth International Computer Assisted Assessment Conference*, Loughborough University, Leicestershire, UK.
- Matveeva, I., Levow, G., Farahat, A., Royer, C., 2005. Generalized latent semantic analysis for term representation. In *Proc. of RANLP*.
- McCulloch, W.S., Pitts, W., 1943. A Logical Calculus of the Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics*. Vol.5, 114-137.
- Mihalcea, R., Corley, C. , Strapparava, C., 2006. Corpus based and knowledge based measures of text semantic similarity. In *Proceedings of the American Association for Artificial Intelligence*.(Boston, MA).
- Miller, G. A., 1986. Dictionaries in the Mind. *Language and cognitive processes*, 1(3), 171-185.
- Miller, G.A., Beckwith, R., Fellbaum, C.D., Gross, D., Miller, K., 1990. WordNet: An online lexical database. *Int. J. Lexicograph.* 3, 4, pp. 235–244.
- Ming, P. Y., Mikhailov, A. A. ,Kuan, T. L., 2000. Intelligent Essay Marking System. In: Cheers, C. (ed.) *Learners Together*. February, NgeeANN Polytechnic, Singapore.
- Mitchell, T., Russell, T., Broomhead, P. Aldridge, N., 2002. Towards Robust Computerised Marking of Free-Text Responses. In: Danson, M. (ed.): *Proceedings of the Sixth International Computer Assisted Assessment Conference*, Loughborough University, Leicestershire, UK.

- Mohler, M., Mihalcea, R., 2009. Text-to-text semantic similarity for automatic short answer grading. In Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (pp. 567-575). Association for Computational Linguistics.
- Mohler, M., Bunescu, R., Mihalcea, R., 2011. Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. Association for Computational Linguistics, 2011, pp 752-762.
- Moore, M.G., 1993. Contemporary Issues In American Distance Education, Elmsford, 135, NY: Pergamon.
- Nabiyev, V.V., 2012. Sayfa 776. Yapay Zeka. Seçkin Yayıncılık, Ankara.
- Needleman, B. S., Wunsch, D. C., 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins, Journal of Molecular Biology 48(3): 443-53.
- Nivens-Bower, C., 2002. Faculty-WritePlacer Plus score comparisons. In Vantage Learning, Establishing WritePlacer Validity: A summary of studies (p 12). (RB-781). Yardley, PA: Author. Norusis, M. J. (2004). SPSS 12.0 guide to data analysis. Upper Saddle River, NJ: Prentice Hall.
- Orhan, Z., 2006. Türkçe Metinlerdeki Anlam Belirsizliği Olan Sözcüklerin Bilgisayar Algoritmaları ile Anlam Belirginleştirilmesi. İstanbul Teknik Üniversitesi. Fen Bilimleri Enstitüsü, Doktora Tezi, 124, İstanbul.
- ÖSYM, 2013. Açık Uçlu Sorularla Deneme Sınavının Uygulanması. <http://www.osym.gov.tr/belge/1-19410/acik-uclu-sorularla-deneme-sinavinin-uygulanmasi-311201-.html>. Son Erişim Tarihi: 10.01.2014
- Özdemir, V., 2009. Word Sense Disambiguation for Turkish Lexical Sample. Fatih University. Institute of Sciences and Engineering, Master Thesis, 63, İstanbul.
- Page, E. B., 1966., Sayfa 238-243. The Imminence of Grading Essays by Computer. Phi Delta Kappan. January.
- Page, E. B., 1968. The use of the computer in analyzing student essays. International Review of Education, 14, 210-225.
- Page, E. B., 1994. Computer Grading of Student Prose, Using Modern Concepts and Software. Journal of Experimental Education, vol. 62, pp. 127-142.

- Page, E. B., 2003. Project essay grade: PEG. In: M. D. Shermis & J. C. Burstein (Eds.), Automated essay scoring: A cross-disciplinary perspective (pp. 43–54). Hillsdale, NJ: Erlbaum.
- Papineni, K., Roukos, S., Ward, T., Zhu, W. 2001. BLEU: A Method for Automatic Evaluation Of Machine Translation. Research report, IBM.
- Per, M.U., 2011. Developing A Concept Extraction System for Turkish. The Middle East Technical University. The Institute of Informatics, Master Thesis, 59, Ankara.
- Pérez, D., Gliozzo, A. M., Strapparava, C., Alfonseca, E., Rodríguez, P., Magnini, B., 2005. Automatic Assessment of Students' Free-Text Answers Underpinned by the Combination of a BLEU-Inspired Algorithm and Latent Semantic Analysis. In FLAIRS Conference (pp. 358-363).
- Peter, K., 2009. Experiments on the difference between semantic similarity and relatedness. In Proceedings of the 17th Nordic Conference on Computational Linguistics - NODALIDA '09, Odense, Denmark.
- Peterson, J. L., 1980. Computer programs for detecting and correcting spelling errors, *Comm. Assoc. Comput. Mach.*, 23:676-687.
- Pulman, S.G., Sukkariéh, J.Z., 2005. Automatic Short Answer Marking. ACL WS Bldg Ed Apps using NLP.
- Rahutomo, F., Aritsugi, M., 2014. Econo-ESA in semantic text similarity. *SpringerPlus*, 2014, 3.1: 149
- Richey, R.C., Klein, J.D. ve Nelson, W.A., 2003, p.1099–1130. Development research: Studies of instructional design and development. D.H. Jonassen (Ed.), *Handbook of research for educational communications and technolog.* Mahwah, NJ: Lawrence Erlbaum Associates.
- Rudner, L., Gagne, P., 2001. An overview of three approaches to scoring written essays by Computer, ERIC Clearinghouse on Assessment and Evaluation, (ERIC Digest number ED 458 290).
- Rudner, L. M., Liang, T., 2002. Automated essay scoring using Bayes' theorem. *The Journal of Technology, Learning, and Assessment*, 1(2).
- Rudner, L., Garcia, V., Welch, C., 2005. An Evaluation of Intellimetric™ Essay Scoring System Using Responses to GMAT® AWA Prompts (GMAC Research report number RR-05-08). <http://www.gmac.com/gmac/researchandtrends/> Son Erişim Tarihi: 10.09.2014

- Rudner, L. M., Garcia, V., Welch, C., 2006. An evaluation of IntelliMetric™ essay scoring system. *The Journal of Technology, Learning and Assessment*, 4 (4).
- Savaşçı, A., 2010. Türkçenin Bitişkenlik Derecesinin İstatistiksel Verilerle Belirlenmesi. Ege Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 58, İzmir.
- Schank, R.C., Abelson, R.P. , 1977. *Scripts, Plans, Goals, and Understanding*, Lawrence Erlbaum, Hillsdale, NJ.
- Shermis, M., Barrera, F., 2002. Exit assessments: Evaluating writing ability through Automated Essay Scoring (ERIC document reproduction service no ED 464 950).
- Shermis, M. D., Raymat, M. V., Barrera, F., 2003. Assessing writing through the curriculum with Automated Essay Scoring (ERIC document reproduction service no ED 477 929).
- Shylov, M., 2008. Turkish and Turkmen Morphological Analyzer and Machine Translation Program. Fatih University, Institute of Sciences and Engineering, Master Thesis, 46, İstanbul.
- Skowron, A., Suraj, Z. (Eds.). 2012. *Rough Sets and Intelligent Systems-Professor Zdzisław Pawlak in Memoriam (Vol. 2)*. Springer Science & Business Media.
- Smith, F. T., Waterman, S. M., 1981. Identification of Common Molecular Subsequences, *Journal of Molecular Biology* 147: 195-197.
- Soysal, E., 2010. *Ontology Based Information Extraction on Free Text Radiological Reports Using Natural Language Processing Approach*. The Middle East Technical University, The Institute of Informatics, Doctorate Thesis, 110, Ankara.
- Stamou, S., Oflazer, K., Pala, K., Christoudoulakis, D., Cristea, D., Tufis, D., Koeva, S., Totkov, G., Dutoit, D., Grigoriadou, M., 2002. Balkanet: A multilingual semantic network for the balkan languages. In *Proceedings of the International Wordnet Conference, Mysore, India* (pp. 21-25).
- Stefanini, M. H., Demazeau, Y., 1995. TALISMAN: A multi-agent system for natural language processing, in *Advances in Artificial Intelligence*, Springer Berlin Heidelberg, pp. 312-322.
- Streeter, L., Psołka, J., Laham, D., MacCuish, D., 2004. The credible grading machine: Essay scoring in the DOD [Department of Defense]. From <http://www.k-a-t.com/papers/essayscoring.pdf>, Son Erişim Tarihi: 10.09.2014

- Şentürk, F., 2009. Biçimbirimsel Bul ve Değiştir. İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 129, İstanbul.
- Tahiroğlu, B.T., 2010. Bilgisayar Destekli Sözlük Bilimi Çalışmalarında Derleme Sözlüğü Veri Tabanı Örneği. Çukurova Üniversitesi. Sosyal Bilimler Enstitüsü, Doktora Tezi, 207, Adana.
- TSPS, 2015. <http://projeler.sdu.edu.tr/tess/>, Erişim Tarihi: 31.10.2015.
- Turna, S.E., 2011. Sözcük Türlerinin Belirlenmesi için Türkçe Kelime Yapılarının Tanımlanması. Dokuz Eylül University, Graduate School of Natural and Applied Sciences, Master Thesis, 38, İzmir.
- Turney, P., 2001. Mining the web for synonyms: PMIIR versus LSA on TOEFL. In Proceedings of the Twelfth European Conference on Machine Learning (ECML).
- Tülek, M., 2007. Türkçe için Metin Özetleme. İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 88, İstanbul.
- Türk Dil Kurumu Sözlük, 2015. <http://www.tdk.gov.tr/>, Türk Dil Kurumu. Erişim Tarihi:21.09.2015.
- Türkçe Eş Anlamlar Sözlüğü, 2015. Mythes-tr, <https://github.com/maidis/mythes-tr>, Türkçe Eş Anlamlar Sözlüğü Erişim Tarihi: 20.08.2015.
- Williams, B., 2012. Essay Grading, Blue Wren Software Pty. Ltd., <http://trial.essaygrading.com>. Son erişim tarihi: 02.02.2014.
- Winkler W. E., 1990. String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage, Proceedings of the Section on Survey Research Methods, American Statistical Association, 354-359.
- Vantage Learning, 2003a. Assessing the accuracy of IntelliMetric for scoring a district-wide writing assessment (RB-806). Newtown, PA: Vantage Learning.
- Vantage Learning, 2003b. How does IntelliMetric score essay responses? (RB 929). Newtown, PA: Vantage Learning.
- Vantage Learning, 2003c. A true score study of 11th grade student writing responses using IntelliMetric Version 9.0 (RB-786). Newtown, PA: Vantage Learning.
- Vantage Learning, 2014. <http://www.vantagelearning.com/products/intellimetric/> Son Erişim Tarihi: 10.09.2014
- Wikipedi, 2015. <https://tr.wikipedia.org/wiki/Wikipedi>, accessed 9 July 2015

- Yıldırım, Ö., 2005. Improved Handling of Sms Messages with statistical Natural Language Processing Techniques. Boğaziçi University, Master Thesis, 66, İstanbul.
- Yıldırım, A., Şimşek, H., 2000. Sayfa 140. Sosyal Bilimlerde Araştırma Yöntemleri. Ankara. Seçkin Yayıncılık.
- Yılmaz, E.Ç., 2008. A Turkish Natural Language Interface for the Semantic Web: A Case Study on Turkish Universities. Atılım University, Graduate School of Natural and Applied Sciences, Master Thesis, 42, Ankara.
- Yılmaz, S., 2009. Türkçe için İyileştirilmiş Biçimbilimsel Çözümleyici. İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, 55, İstanbul.
- Zafer, H.R., 2011. A Generic Syntactic Parser for Turkic Languages. Fatih University, Institute of Sciences and Engineering, Master Thesis, 59, İstanbul.
- Zhang, Y., Witte, R., Rilling, J., Haarslev, V., 2008. 'Ontological approach for the semantic recovery of traceability links between software artefacts'. IET software, 2008, 2(3), 185-203

EKLER

EK A. Metin Araçları Sınıfı

EK B. Türkçe Gereksiz Kelimeler

EK C. Kök Bulucu

EK D. Eş Anlam Bulucu

EK E. Yakın Anlam Bulucu

EK F. Kosinüs Benzerliği

EK G. Kısaltma Bulucu Sınıfı

EK H. Rakamların Yazıya Çevrilmesi

EK I. GAA Fonksiyonu

EK İ. TGAA Fonksiyonu

EK J. Bilgisayar Ağları Deneme Sınavı

EK K. Hata Denetimi Kodları

EK L. Levestein Düzenleme Uzaklık Algoritması

EK M. Birinci Cevap Anahtarına Göre Kosinüs Benzerlik, GAA ve TGAA Puanları

EK N. İkinci Cevap Anahtarına Göre Kosinüs Benzerlik, GAA ve TGAA Puanları

EK O. Eğitimci1 Cevap Anahtarı

EK Ö. Eğitimci2 Cevap Anahtarı

EK P. TSPS Puan ile Eğitimci1 Puan Pearson t Testi Sonuçları

EK R. TSPS Puan ile Eğitimci2 Puan Pearson t Testi Sonuçları

EK S. TSPSORT Puan ile Eğitimci1 Puan Pearson t Testi Sonuçları

EK Ş. TSPSORT Puan ile Eğitimci2 Puan Pearson t Testi Sonuçları

EK A. Metin Araçları Sınıfı

```
public static class MetinAraclari
{
    public static string KelimeHTMLTemizle(string Kelime)
    {
        string temizKelime = Regex.Replace(Kelime, @"<(.|\n)*?>", " ");
        return temizKelime;
    }

    public static string KelimeFiltreleKoksuz(string pMetin)
    {
        //Boşlukların kırılması
        pMetin = pMetin.Trim();

        //Hepsinin küçük harfe dönüştürülmesi
        pMetin = pMetin.ToLower();

        //Noktalama işaretlerinin kaldırılması
        pMetin = Regex.Replace(pMetin, @"[^\w]", " ");
        return pMetin;
    }

    public static string KelimeFiltrele(string pMetin)
    {
        //Boşlukların kırılması
        pMetin = pMetin.Trim();

        //Hepsinin küçük harfe dönüştürülmesi
        pMetin = pMetin.ToLower();

        //Noktalama işaretlerinin kaldırılması
        pMetin = Regex.Replace(pMetin, @"[^\w]", " ");

        //Kelime köklerinin bulunması
        string[] kelimeler = pMetin.Split(' ');
        List<string> KelimeListesi = new List<string>();
        string[] metin = null;
        if (HttpRuntime.Cache["Gereksizler"] == null)
        {
            //Gereksiz kelimelerin kaldırılması
            string sr =
                File.ReadAllText(AppDomain.CurrentDomain.BaseDirectory.ToString() +
                    "\\Old_App_Code\\Base\\stopwords.txt", System.Text.Encoding.GetEncoding("iso-8859-15"));
            metin = sr.Split('\n');
            HttpRuntime.Cache.Add("Gereksizler", metin, null,
                DateTime.Now.AddMinutes(60), System.Web.Caching.Cache.NoSlidingExpiration,
                System.Web.Caching.CacheItemPriority.Normal, null);
        }
        else
        {
            metin = HttpRuntime.Cache["Gereksizler"] as string[];
        }
        foreach (string kelime in kelimeler)
        {
            bool eklenecek = true;
            try
            {
                foreach (string item in metin)
                {
                    if (item.IndexOf('\r') > 0)

```

```

        {
            string s = item.Remove(item.IndexOf('\r'), 1);
            if (kelime == s)
                eklenecek = false;
        }
    }
}
catch (Exception ex)
{
    throw ex;
}
if (!string.IsNullOrEmpty(kelime) && eklenecek == true)
    KelimeListesi.Add(kelime.Trim());
}

//Burada sayı varsa bunların yazıya çevirilmesi
for (int i = 0; i < KelimeListesi.Count; i++)
{
    long sonuc = 0;
    if (long.TryParse(KelimeListesi[i], out sonuc))
    {
        KelimeListesi[i] = RakamiYaziyaCevir.YaziyaCevirTR(sonuc);
    }
}

//Kelime listesindeki her bir kelimenin kökünün bulunması
List<string> KelimeKokListesi = new List<string>();
KokBulucu kokBulucu = new KokBulucu();
foreach (var item in KelimeListesi)
{
    try
    {
        string sonuc = kokBulucu.KokBul(item);
        KelimeKokListesi.Add(sonuc.Trim());
    }
    catch (Exception ex)
    {
        throw ex;
    }
}

//Kökleri bulunan kelimelerin birleştirilmesi
pMetin = string.Empty;
foreach (var item in KelimeKokListesi)
{
    if (string.IsNullOrEmpty(pMetin))
        pMetin = item;
    else
        pMetin += " " + item;
}

//Birleştirilen ve filtreden geçen metnin geri döndürülmesi
return pMetin;
}
}

```

EK B. Türkçe Gereksiz kelimeler

acaba	mu
ama	mü
bana	mı
bazı	nasıl
belki	ne
ben	neden
benden	nerde
beni	nerede
benim	nereye
biri	niye
birkaç	niçin
birkez	on
birşey	ona
birşeyi	ondan
biz	onlar
bizden	onlardan
bizi	onları
bizim	onların
bu	onu
buna	otuz
bunda	sanki
bundan	seksen
bunu	sen
bunun	senden
da	seni
daha	senin
dahi	siz
de	sizden
defa	sizi
diye	sizin
en	tüm
gibi	ve
hem	veya
hep	ya
hepsi	yani
her	çok
hiç	çünkü
iki	şey
ile	şeyden
ise	şeyi
için	şeyler
kez	şu
ki	şuna
kim	şunda
kimden	şundan
kime	şunu
kimi	

EK C. Kök Bulucu

```
public class KokBulucu
{
    public KokBulucu()
    {
    }

    Zemberek z = new met.zemberek.erisim.Zemberek(new TurkiyeTurkcesi());
    public string KokBul(string item)
    {
        KokBulucuUretici kob = z.dilBilgisi().kokler().getKokBulucuFactory();
        List<met.zemberek.yapi.Kok> l =
        kob.getKesinKokBulucu().getAdayKokler(item);
        string sonuc = string.Empty;
        if (l != null && l.Count > 0)
        {
            sonuc = l[l.Count - 1].icerik();
        }
        else
            sonuc = item;
        return sonuc;
    }
}
```

EK D. Eş Anlam Bulucu

```
public static class EsAnlamBulucu
{
    public static List<string> EsAnlamliListesi(string pAranacakKelime)
    {
        List<string> esAnlamlilarListem = new List<string>();
        try
        {
            List<Kelime> KelimeListesi = new List<Kelime>();
            if (HttpContext.Cache["EsAnlamlilar"] == null)
            {
                StreamReader stream = new
                StreamReader(AppDomain.CurrentDomain.BaseDirectory.ToString() +
                "\\Old_App_Code\\Base\\th_tr_TR_v2.dat", new UTF8Encoding(true, true));
                string tumVeri = stream.ReadToEnd();
                stream.Close();
                string[] satirlar = tumVeri.Split('\n');
                for (int i = 0; i < satirlar.Length; i++)
                {
                    string[] kelimeler = satirlar[i].Split('|');
                    int anlamSayisi = 0;
                    if (kelimeler.Length == 2 && Int32.TryParse(kelimeler[1],
                    out anlamSayisi) && anlamSayisi < 50)
                    {
                        string[] icDizi = new string[anlamSayisi];
                        for (int j = 0; j < anlamSayisi; j++)
                        {
                            icDizi[j] = satirlar[1 + i + j];
                        }
                        for (int k = 0; k < icDizi.Length; k++)
                        {
                            string[] enIcDizi = icDizi[k].Split('|');
                            string tur = enIcDizi[0];
                            enIcDizi = enIcDizi.Where(val => val !=
                            enIcDizi[0]).ToArray();
                            Kelime kelime = new Kelime(kelimeler[0], tur,
                            KelimeListesi.Add(kelime);
                        }
                    }
                }
                HttpContext.Cache.Add("EsAnlamlilar", KelimeListesi, null,
                DateTime.Now.AddMinutes(60), System.Web.Caching.Cache.NoSlidingExpiration,
                System.Web.Caching.CacheItemPriority.Normal, null);
            }
            else
            {
                KelimeListesi = HttpContext.Cache["EsAnlamlilar"] as
                List<Kelime>;
            }

            if (KelimeListesi.Count > 0)
            {
                var liste = from kelime in KelimeListesi where
                kelime.ToString() == pAranacakKelime select kelime;
                foreach (Kelime bulunan in liste)
                {
                    foreach (string anlam in bulunan.anlamlar)
                    {

```

```
        esAnlamlilarListem.Add(anlam);
    }
}
return esAnlamlilarListem;
}
catch (IOException)
{
    return esAnlamlilarListem;
}
}
```



EK E. Yakın Anlam Bulucu

```
public static class YakınAnlamBulucu
{
    public static List<string> YakınAnlamListesi(string pAranacakKelime)
    {
        List<string> yakınAnlamlılarListem = new List<string>();
        try
        {
            List<Iliski> KelimeListesi = new List<Iliski>();
            if (HttpContext.Cache["YakınAnlamlılar"] == null)
            {
                try
                {
                    string tumVeri =
File.ReadAllText(AppDomain.CurrentDomain.BaseDirectory.ToString() +
"\Old_App_Code\Base\relations.txt", Encoding.Default);
                    string[] satirlar = tumVeri.Split('\n');
                    foreach (string satir in satirlar)
                    {
                        string[] elemanlar = satir.Split('\t');
                        Iliski kelime = new Iliski();
                        kelime.Kavram1 = elemanlar[0];
                        kelime.Kavram2 = elemanlar[2];
                        kelime.IliskiTur = elemanlar[1];
                        KelimeListesi.Add(kelime);
                    }
                    HttpContext.Cache.Add("YakınAnlamlılar", KelimeListesi,
null, DateTime.Now.AddMinutes(60),
System.Web.Caching.Cache.NoSlidingExpiration,
System.Web.Caching.CacheItemPriority.Normal, null);
                }
                catch (IOException)
                {
                }
            }
            else
            {
                KelimeListesi = HttpContext.Cache["YakınAnlamlılar"] as
List<Iliski>;
            }

            if (KelimeListesi.Count > 0)
            {
                var liste = from kelime in KelimeListesi where kelime.Kavram1
== pAranacakKelime || kelime.Kavram2 == pAranacakKelime select kelime;
                foreach (Iliski bulunan in liste)
                {
                    if (bulunan.IliskiTur == "HYPERNYMY" &&
!yakınAnlamlılarListem.Contains(bulunan.Kavram1) && bulunan.Kavram1 !=
pAranacakKelime)
                        yakınAnlamlılarListem.Add(MetinAraclari.KelimeFiltrele(bulunan.Kavram1));
                    if (bulunan.IliskiTur == "HYPERNYMY" &&
!yakınAnlamlılarListem.Contains(bulunan.Kavram2) && bulunan.Kavram2 !=
pAranacakKelime)
                        yakınAnlamlılarListem.Add(MetinAraclari.KelimeFiltrele(bulunan.Kavram2));
                }
            }
            return yakınAnlamlılarListem;
        }
    }
}
```

```
    }  
    catch (IOException)  
    {  
        return yakinAnlamlilarListem;  
    }  
}
```



EK F. Kosinüs Benzerliği

```
public static Double COSINE(string pOgrenciCevap, string pHocaCevap)
{
    pOgrenciCevap = MetinAraclari.KelimeFiltrele(pOgrenciCevap);
    pHocaCevap = MetinAraclari.KelimeFiltrele(pHocaCevap);
    string[] giris = pOgrenciCevap.Split(' ');
    string[] CevapAnahtari = pHocaCevap.Split(' ');

    //Hoca cevabındaki kelimenin eş anlamlı listesinde öğrencinin cevabı
    varsa öğrenci cevabı hocanınkine eşit olacak
    foreach (string cevap in CevapAnahtari)
    {
        for (int i = 0; i < giris.Length; i++)
        {
            if (EsAnlamBulucu.EsAnlamliListesi(cevap).Contains(giris[i]))
            {
                giris[i] = cevap;
            }
        }
    }

    List<string> EsasKelimeler = new List<string>();

    foreach (string item in giris)
    {
        if (!EsasKelimeler.Contains(item))
            EsasKelimeler.Add(item);
    }
    foreach (string item in CevapAnahtari)
    {
        if (!EsasKelimeler.Contains(item))
            EsasKelimeler.Add(item);
    }

    int[] vektor1 = new int[EsasKelimeler.Count];
    for (int i = 0; i < EsasKelimeler.Count; i++)
    {
        int sayac = 0;
        foreach (var item in giris)
        {
            if (item == EsasKelimeler[i])
                sayac++;
        }
        vektor1[i] = sayac;
    }

    int[] vektor2 = new int[EsasKelimeler.Count];
    for (int i = 0; i < EsasKelimeler.Count; i++)
    {
        int sayac = 0;
        foreach (var item in CevapAnahtari)
        {
            if (item == EsasKelimeler[i])
                sayac++;
        }
        vektor2[i] = sayac;
    }

    int dot = 0;
```

```
for (int i = 0; i < vektor1.Length; i++)
{
    dot += vektor1[i] * vektor2[i];
}

double toplamVektor1 = 0;
for (int i = 0; i < vektor1.Length; i++)
{
    toplamVektor1 += Math.Pow(vektor1[i], 2);
}
double d1 = Math.Sqrt(toplamVektor1);

double toplamVektor2 = 0;
for (int i = 0; i < vektor2.Length; i++)
{
    toplamVektor2 += Math.Pow(vektor2[i], 2);
}
double d2 = Math.Sqrt(toplamVektor2);

return dot / (d1 * d2);
}
```

EK G. Kısaltma Bulucu Sınıfı

```
public static class KısaltmaBulucu
{
    public static string KısaltmaKarsiligi(string pAranacakKelime)
    {
        string kısıltmaKarsiligi = pAranacakKelime; ;
        try
        {
            //cachleme yap

            List<Kısaltma> KelimeListesi = new List<Kısaltma>();
            try
            {
                if (HttpContext.Cache["Kısaltmalar"] == null)
                {
                    string tumVeri =
                    File.ReadAllText(AppDomain.CurrentDomain.BaseDirectory.ToString() +
                    "\\Old_App_Code\\Base\\kısaltmalar.txt", Encoding.Default);
                    string[] satirlar = tumVeri.Split('\n');
                    foreach (string satir in satirlar)
                    {
                        string[] elemanlar = satir.Split('\t');
                        Kısaltma kelime = new Kısaltma();
                        kelime.Kisa = elemanlar[0];
                        kelime.Uzun = elemanlar[1];
                        KelimeListesi.Add(kelime);
                    }
                    HttpContext.Cache.Add("Kısaltmalar", KelimeListesi, null,
                    DateTime.Now.AddMinutes(60), System.Web.Caching.Cache.NoSlidingExpiration,
                    System.Web.Caching.CacheItemPriority.Normal, null);
                }
                else
                {
                    KelimeListesi = HttpContext.Cache["Kısaltmalar"] as
                    List<Kısaltma>;
                }
            }
            catch (IOException)
            {
            }

            if (KelimeListesi.Count > 0)
            {
                var liste = from kelime in KelimeListesi where kelime.Kisa ==
                pAranacakKelime select kelime;
                foreach (var item in liste)
                {
                    kısıltmaKarsiligi =
                    MetinAraclari.KelimeFiltrele(item.Uzun);
                    break;
                }
            }
            return kısıltmaKarsiligi;
        }
        catch (IOException)
        {
            return kısıltmaKarsiligi;
        }
    }
}
```

EK H. Rakamların Yazıya Çevrilmesi

```
public static class RakamiYaziyaCevir
{
    public static string YaziyaCevirTR(int _iSayi, string _sAyrac)
    {
        if (_sAyrac == null ||
        _iSayi.ToString().IndexOf(_sAyrac.ToCharArray()[0]) == -1)
        {
            return YaziyaCevirTR(_iSayi);
        }

        string[] sSayilar =
        _iSayi.ToString().Split(_sAyrac.ToCharArray());
        return YaziyaCevirTR(Convert.ToInt32(sSayilar[0])) +
        YaziyaCevirTR(Convert.ToInt32(sSayilar[1]));
    }

    public static string YaziyaCevirTR(long _iSayi)
    {
        if(_iSayi==0)
        {
            return "sıfır";
        }

        string[] Birler = { "", " Bir", " İki", " Üç", " Dört", " Beş", "
Altı", " Yedi", " Sekiz", " Dokuz" };
        string[] Onlar = { "", " On", " Yirmi", " Otuz", " Kırk", " Elli",
" Altmış", " Yetmiş", " Seksen", " Doksan" };
        string[] Yuzler = { "", " Yüz", " İkiYüz", " ÜçYüz", " DörtYüz", "
BeşYüz", " AltıYüz", " YediYüz", " SekizYüz", " DokuzYüz" };
        string[] Binler = { " Trilyon", " Milyar", " Milyon", " Bin", ""
};

        string[] Dizi = new string[15];
        string[] Uc_hane = new string[3];
        string Yazı = "";
        string Sonuc = "";

        string Sayi_str;
        Sayi_str = _iSayi.ToString();
        Sayi_str = "00000000000000" + Sayi_str;
        Sayi_str = Sayi_str.Substring(_iSayi.ToString().Length, 15);

        for (int i = 0; i < Sayi_str.Length; i++)
        {
            Dizi[i] = Sayi_str[i].ToString();
        }

        for (int i = 0; i < 5; i++)
        {
            Uc_hane[0] = Dizi[(i * 3) + 0];
            Uc_hane[1] = Dizi[(i * 3) + 1];
            Uc_hane[2] = Dizi[(i * 3) + 2];

            if (Uc_hane[0] == "0")
            {
                Yazı = "";
            }
        }
    }
}
```

```

else if (Uc_hane[0] == "1")
{
    Yazı = "Yüz";
}
else
{
    Yazı = Birler[Convert.ToInt16(Uc_hane[0])] + " Yüz";
}

Yazı = Yazı + Onlar[Convert.ToInt16(Uc_hane[1])] +
Birler[Convert.ToInt16(Uc_hane[2])];

if (Yazı != "")
    Yazı = Yazı + " " + Binler[i];

if ((i == 2) && (Yazı == "BirBin"))
{
    Yazı = "Bin";
}

Sonuc += Yazı;
}

Sonuc = Sonuc.ToLower();
return Sonuc.Trim();
}
}

```

EK I. GAA Fonksiyonu

```
public static SoruPuanCevap[] LSA(string pHocaCevap, SoruPuanCevap[]
pOgrenciCevapListesi)
{
    //İşleme başla
    List<string> KelimelerSatiri = new List<string>();
    //Filtreleme yap
    pHocaCevap = MetinAraclari.KelimeFiltrele(pHocaCevap);
    string[] CevapAnahtari = pHocaCevap.Split(' ');

    //ÖğrenciCevaplarının ayrıştırılması
    List<string[]> ogrenciKisiselCevapListesi = new List<string[]>();
    foreach (SoruPuanCevap item in pOgrenciCevapListesi)
    {
        //Filtreleme yap
        string cevap = MetinAraclari.KelimeFiltrele(item.Cevap);
        ogrenciKisiselCevapListesi.Add(cevap.Split(' '));
    }
    ogrenciKisiselCevapListesi.Add(CevapAnahtari);
    foreach (string[] cevaplar in ogrenciKisiselCevapListesi)
    {
        foreach (var item in cevaplar)
        {
            if (!KelimelerSatiri.Contains(item))
                KelimelerSatiri.Add(item);
        }
    }
    ogrenciKisiselCevapListesi.RemoveAt(ogrenciKisiselCevapListesi.Count -
1);

    KelimelerSatiri.Sort();

    int M = KelimelerSatiri.Count();
    int N = pOgrenciCevapListesi.Length;

    List<int[]> VektorListesi = new List<int[]>(N);
    foreach (var item in ogrenciKisiselCevapListesi)
    {
        int[] vektor = new int[KelimelerSatiri.Count];
        for (int i = 0; i < KelimelerSatiri.Count; i++)
        {
            int sayac = 0;
            foreach (var kelime in item)
            {
                if (kelime == KelimelerSatiri[i])
                    sayac++;
            }
            vektor[i] = sayac;
        }
        VektorListesi.Add(vektor);
    }

    double[,] X = new double[M, N];

    for (int i = 0; i < M; i++)
```

```

{
    for (int j = 0; j < N; j++)
    {
        X[i, j] = VektorListesi[j][i];
    }
}

SingularValueDecomposition svd = null;
double[,] U = null;
double[,] S = null;
double[,] V = null;
if (M < N)
{
    svd = new SingularValueDecomposition(X, true, true, true);
    U = svd.LeftSingularVectors;
    S = svd.DiagonalMatrix;
    V = svd.RightSingularVectors;
}
else
{
    svd = new SingularValueDecomposition(X, true, true, false);
    U = svd.LeftSingularVectors;
    S = svd.DiagonalMatrix;
    V = svd.RightSingularVectors;
}

//q denklemini çalıştırılacak

double[,] vektor4 = new double[1, KelimelerSatiri.Count];
for (int i = 0; i < KelimelerSatiri.Count; i++)
{
    int sayac = 0;
    foreach (var item in CevapAnahtari)
    {
        if (item == KelimelerSatiri[i])
            sayac++;
    }
    vektor4[0, i] = sayac;
}

int k = 2;

double[,] Uk = new double[KelimelerSatiri.Count, k];
for (int i = 0; i < M; i++)
{
    for (int j = 0; j < k; j++)
    {
        Uk[i, j] = U[i, j];
    }
}

double[,] SkTers = new double[k, k];
for (int i = 0; i < k; i++)
{
    for (int j = 0; j < k; j++)
    {
        SkTers[i, j] = (S[i, j] == 0) ? 0 : 1 / S[i, j];
    }
}

double[,] C = new double[1, k];

```

```

for (int i = 0; i < 1; i++)
{
    for (int j = 0; j < k; j++)
    {
        for (int a = 0; a < KelimelerSatiri.Count; a++)
        {
            C[i, j] += vektor4[i, a] * Uk[a, j];
        }
    }
}

double[,] Q = new double[1, k];
for (int i = 0; i < 1; i++)
{
    for (int j = 0; j < k; j++)
    {
        for (int a = 0; a < k; a++)
        {
            Q[i, j] += C[i, a] * SkTers[a, j];
        }
    }
}

// Cosine Uygulaması

double[,] Vk = new double[N, k];
for (int i = 0; i < N; i++)
{
    for (int j = 0; j < k; j++)
    {
        Vk[i, j] = V[i, j];
    }
}

//Ogrencilerin Cevaplarının değerlendirilmesi
for (int i = 0; i < pOgrenciCevapListesi.Length; i++)
{
    double sonuc = ((Vk[i, 0] * Q[0, 0]) + (Vk[i, 1] * Q[0, 1])) /
    Math.Abs(((Math.Sqrt(Math.Pow(Q[0, 0], 2) + Math.Pow(Q[0, 1], 2)))) *
    Math.Abs((Math.Sqrt(Math.Pow(Vk[i, 0], 2) + Math.Pow(Vk[i, 1], 2))))));
    if (sonuc.ToString() == "NaN")
        sonuc = 0;
    pOgrenciCevapListesi[i].Puan = Math.Abs(sonuc);
}

return pOgrenciCevapListesi;
}

```


EK İ. TGAA Fonksiyonu

```
public static SoruPuanCevap[] TGAA(string pHocaCevap, SoruPuanCevap[]
pOgrenciCevapListesi)
{
    try
    {
        //İşleme başla
        List<string> KelimelerSatiri = new List<string>();
        //Filtreleme yap
        pHocaCevap = MetinAraclari.KelimeFiltrele(pHocaCevap);
        string[] CevapAnahtari = pHocaCevap.Split(' ');

        //ÖğrenciCevaplarının ayrıştırılması
        List<string[]> ogrenciKisiselCevapListesi = new List<string[]>();
        foreach (SoruPuanCevap item in pOgrenciCevapListesi)
        {
            //Filtreleme yap
            string cevap = MetinAraclari.KelimeFiltrele(item.Cevap);
            ogrenciKisiselCevapListesi.Add(cevap.Split(' '));
        }

        //Hoca cevabındaki kelimenin eş anlamlı listesinde öğrencinin
        cevabı varsa öğrenci cevabı hocanıninkine eşit olacak
        foreach (string cevap in CevapAnahtari)
        {
            for (int i = 0; i < ogrenciKisiselCevapListesi.Count; i++)
            {
                for (int j = 0; j < ogrenciKisiselCevapListesi[i].Length;
                j++)
                {
                    if
                    (EsAnlamBulucu.EsAnlamliListesi(cevap).Contains(ogrenciKisiselCevapListesi[i][
                    j]))
                    {
                        ogrenciKisiselCevapListesi[i][j] = cevap;
                    }
                }
            }
        }

        //Hoca cevabındaki kelimenin yakın anlamlı listesinde öğrencinin
        cevabı varsa öğrenci cevabı hocanıninkine eşit olacak
        foreach (string cevap in CevapAnahtari)
        {
            for (int i = 0; i < ogrenciKisiselCevapListesi.Count; i++)
            {
                for (int j = 0; j < ogrenciKisiselCevapListesi[i].Length;
                j++)
                {
                    if
                    (YakinAnlamBulucu.YakinAnlamListesi(cevap).Contains(ogrenciKisiselCevapListesi
                    [i][j]))
                    {
                        ogrenciKisiselCevapListesi[i][j] = "|" + cevap;
                    }
                }
            }
        }
    }
}
```

```

    }
}

ogrenciKisiselCevapListesi.Add(CevapAnahtari);
foreach (string[] cevaplar in ogrenciKisiselCevapListesi)
{
    foreach (string item in cevaplar)
    {
        string[] icdizi = item.Split(' ');
        foreach (var ic in icdizi)
        {
            if (!KelimelerSatiri.Contains(ic) && ic[0] != '|')
                KelimelerSatiri.Add(ic);
        }
    }
}

```

```
ogrenciKisiselCevapListesi.RemoveAt(ogrenciKisiselCevapListesi.Count - 1);
```

```

KelimelerSatiri.Sort();

int M = KelimelerSatiri.Count();
int N = pOgrenciCevapListesi.Length;

List<double[]> VektorListesi = new List<double[]>(N);
foreach (var item in ogrenciKisiselCevapListesi)
{
    double[] vektor = new double[KelimelerSatiri.Count];
    for (int i = 0; i < KelimelerSatiri.Count; i++)
    {
        double sayac = 0;
        foreach (var kelime in item)
        {
            string[] icdizi = kelime.Split(' ');
            foreach (var ic in icdizi)
            {
                if (KelimelerSatiri[i] == ic)
                    sayac += 1;
                else
                {
                    string ara = ic.Remove(0, 1);
                    if (KelimelerSatiri[i] == ara)
                        sayac += 0.5;
                }
            }
        }
        vektor[i] = sayac;
    }
    VektorListesi.Add(vektor);
}

```

```
double[,] X = new double[M, N];
```

```

for (int i = 0; i < M; i++)
{
    for (int j = 0; j < N; j++)

```

```

        {
            X[i, j] = VektorListesi[j][i];
        }
    }

    SingularValueDecomposition svd = null;
    double[,] U = null;
    double[,] S = null;
    double[,] V = null;
    if (M < N)
    {
        svd = new SingularValueDecomposition(X, true, true, true);
        U = svd.LeftSingularVectors;
        S = svd.DiagonalMatrix;
        V = svd.RightSingularVectors;
    }
    else
    {
        svd = new SingularValueDecomposition(X, true, true, false);
        U = svd.LeftSingularVectors;
        S = svd.DiagonalMatrix;
        V = svd.RightSingularVectors;
    }

    //q denklemini çalıştırılacak

    List<string> iccevaplar = new List<string>();
    foreach (string item in CevapAnahtari)
    {
        string[] icdizi = item.Split(' ');
        foreach (var icitem in icdizi)
        {
            iccevaplar.Add(icitem);
        }
    }
    CevapAnahtari = iccevaplar.ToArray();

    double[,] vektor4 = new double[1, KelimelerSatiri.Count];
    for (int i = 0; i < KelimelerSatiri.Count; i++)
    {
        int sayac = 0;
        foreach (var item in CevapAnahtari)
        {
            if (item == KelimelerSatiri[i])
                sayac++;
        }
        vektor4[0, i] = sayac;
    }

    int k = 2;

    double[,] Uk = new double[KelimelerSatiri.Count, k];
    for (int i = 0; i < M; i++)
    {
        for (int j = 0; j < k; j++)
        {
            Uk[i, j] = U[i, j];
        }
    }

```

```

double[,] SkTers = new double[k, k];
for (int i = 0; i < k; i++)
{
    for (int j = 0; j < k; j++)
    {
        SkTers[i, j] = (S[i, j] == 0) ? 0 : 1 / S[i, j];
    }
}

double[,] C = new double[1, k];
for (int i = 0; i < 1; i++)
{
    for (int j = 0; j < k; j++)
    {
        for (int a = 0; a < KelimelerSatiri.Count; a++)
        {
            C[i, j] += (vektor4[i, a] * Uk[a, j]);
        }
    }
}

double[,] Q = new double[1, k];
for (int i = 0; i < 1; i++)
{
    for (int j = 0; j < k; j++)
    {
        for (int a = 0; a < k; a++)
        {
            Q[i, j] += C[i, a] * SkTers[a, j];
        }
    }
}

// Cosine Uygulaması

double[,] Vk = new double[N, k];
for (int i = 0; i < N; i++)
{
    for (int j = 0; j < k; j++)
    {
        Vk[i, j] = V[i, j];
    }
}

//Ogrencilerin Cevaplarının değerlendirilmesi
for (int i = 0; i < pOgrenciCevapListesi.Length; i++)
{
    double sonuc = ((Vk[i, 0] * Q[0, 0]) + (Vk[i, 1] * Q[0, 1])) /
    ((Math.Sqrt(Math.Pow(Q[0, 0], 2) + Math.Pow(Q[0, 1], 2))) *
    (Math.Sqrt(Math.Pow(Vk[i, 0], 2) + Math.Pow(Vk[i, 1], 2))));
    if (sonuc.ToString() == "NaN")
        sonuc = 0;
    pOgrenciCevapListesi[i].Puan = Math.Abs(sonuc);
}

return pOgrenciCevapListesi;
}

```

```
    catch (Exception ex)
    {
        throw ex;
    }
}
```



EK J. Bilgisayar Ağları Deneme Sınavı

1. OSI referans modelinin birinci katmanındaki cihazları yazınız
2. Ağ topoloji türlerini yazınız
3. OSI referans modelindeki veri iletim katmanının özelliklerini yazınız
4. Doğrusal Topolojinin avantajlarını yazınız
5. Topoloji nedir?
6. Ağda kullanılan veri adlarını yazınız
7. Fiber optik çeşitlerini yazınız
8. Yineleyici nedir?
9. Ağın avantajları nelerdir?
10. OSI referans modelindeki ağ (network) katmanının özelliklerini yazınız



EK K. Hata Denetimi Kodları

```
public class HataDenetimi
{
    Zemberek zemberek = new Zemberek(new TurkiyeTurkcesi());
    public List<string> KelimeDenetle(string Kelime)
    {
        List<string> SonListe = new List<string>();
        if (Kelime.Length > 2) //burada 2 harfden uzun kelimelerin hata
denetimi yapılır
        {
            if (zemberek.kelimeDenetle(Kelime))
            {
                SonListe.Add(Kelime);
            }
            else
            {
                List<string> KelimeListesi = new List<string>();
                string str = string.Empty;
                int sayac = 0;
                for (int j = 0; j < Kelime.Length; j++)
                {
                    for (int i = j; i < Kelime.Length; i++)
                    {
                        str += Kelime[i];
                        if (str.Length > Kelime.Length - 3) //Kelime
uzunluğuna en yakın n-gramlar
                            KelimeListesi.Add(str);
                            sayac++;
                    }
                    if (sayac == Kelime.Length - j)
                    {
                        str = string.Empty;
                        sayac = 0;
                    }
                }

                List<string> OneriListesi = new List<string>();
                foreach (var kelime in KelimeListesi)
                {
                    string[] oneriler = zemberek.oner(kelime);
                    foreach (var oneri in oneriler)
                    {
                        OneriListesi.Add(oneri);
                    }
                }

                List<Oneri> SiraliOneriListesi = new List<Oneri>();
                var oneriList = OneriListesi.GroupBy(u => u).ToList();
                foreach (var oneri in oneriList)
                {
                    var count = OneriListesi.Where(oneriKelime => oneriKelime
== oneri.Key).Count();
                    Oneri o = new Oneri(oneri.Key);
                    o.Frekans = count;
                    SiraliOneriListesi.Add(o);
                }

                List<string> Liste = new List<string>();
                foreach (var so in SiraliOneriListesi)
                {
```

```
        int uzaklik = Levenstein.Uzaklik(Kelime, so.Kelime);
        if (uzaklik > 0 && uzaklik <= 2) // Buradaki 2 threshold
(Eşik) değeri yani kaç tane hata yapılabileceği
        {
            Liste.Add(so.Kelime);
        }
    }

    var sorted = from s in Liste
                 orderby s.Length descending
                 select s;

    foreach (var bulunan in sorted)
    {
        SonListe.Add(bulunan);
    }
}
if (SonListe.Count == 0)
    SonListe.Add(Kelime);
return SonListe;
}
```


EK L. Levestein Düzenleme Uzaklık Algoritması

```
public class Levenstein
{
    public static int Uzaklik(string Source, string Target)
    {
        int[,] matrix3 = new int[Source.Length, Target.Length];
        int distance3 = Source.FindLevenshteinDistance(Target, out
matrix3);
        return distance3;
    }

    public static class StringExtensions
    {
        // Genişletme metodu, karşılaştırma matrisini de out parametresi
olarak döndürmektedir.
        public static int FindLevenshteinDistance(this string Source, string
Target, out int[,] Matrix)
        {
            int n = Source.Length;
            int m = Target.Length;

            Matrix = new int[n + 1, m + 1]; // Hesaplama matrisi üretilir. 2
boyutlu matrisin boyut uzunlukları ise kaynak ve hedef metinlerin karakter
uzunluklarına göre set edilir

            if (n == 0) // Eğer kaynak metin yoksa zaten hedef metnin tüm
harflerinin değişimi söz konusu olduğundan, hedef metnin uzunluğu kadar bir
yakınlık değeri mümkün olabilir
                return m;

            if (m == 0) // Yukarıdaki durum hedefin karakter içermemesi
halinde de geçerlidir
                return n;

            // Aşağıdaki iki döngü ile yatay ve dikey eksenlerdeki standart
0,1,2,3,4...n elemanları doldurulur
            for (int i = 0; i <= n; i++)
                Matrix[i, 0] = i;

            for (int j = 0; j <= m; j++)
                Matrix[0, j] = j;

            // Kıyaslama ve derecelendirme operasyonu yapılır
            for (int i = 1; i <= n; i++)
                for (int j = 1; j <= m; j++)
                {
                    int cost = (Target[j - 1] == Source[i - 1]) ? 0 : 1;
                    Matrix[i, j] = Math.Min(Math.Min(Matrix[i - 1, j] + 1,
Matrix[i, j - 1] + 1), Matrix[i - 1, j - 1] + cost);
                }

            return Matrix[n, m]; // sağ alt taraftaki hücre değeri döndürülür
        }
    }
}
```

EK M. Birinci Cevap Anahtarına Göre Kosinüs Benzerlik, GAA ve TGAA Puanları

Birinci Cevap Anahtarına Göre Kosinüs Benzerliği												
Sıra No	Ad Soyad	S1P	S2P	S3P	S4P	S5P	S6P	S7P	S8P	S9P	S10P	Toplam Puan
1	ogr1	0	2,11	2,38	2,14	2,45	0,83	1,29	0,69	0,4	2,55	14,84
2	ogr2	5,35	4,08	2,98	4,04	1,41	2,24	0	0,96	8,73	4,02	33,81
3	ogr3	5,07	4,08	2,2	2,83	3,77	3,65	0	0,93	1,33	0	23,86
4	ogr4	8,02	4,08	4,27	2,77	3,16	3,16	0	1,52	4,95	3,32	35,25
5	ogr5	4,63	9,85	1,65	1,05	7,67	9,49	0	2,24	1,94	3,79	42,31
6	ogr6	5,35	10	4,52	5,39	8,01	3,16	5,48	0,93	9,81	3,16	55,81
7	ogr7	3,78	10	1,65	3,2	3,3	7,75	5,77	0	0	2,86	38,31
8	ogr8	4,36	9,8	1,65	0	2,92	2,24	0	0,46	1,15	2,12	24,7
9	ogr9	1,69	4,08	2,05	1,81	5,56	4,47	0	3,32	4,9	5,67	33,55
10	ogr10	4,63	10	4,57	5,88	4,3	5,16	2,78	1,02	0,76	2,31	41,41
11	ogr11	6,3	4,08	5,38	4,92	7,67	8,94	0	0	10	5,25	52,54
12	ogr12	6,17	4,08	3,97	2,66	6,77	9,13	0	0,56	9,17	6,26	48,77
13	ogr13	5,67	4,08	5,52	1,85	6,76	4,47	3,54	2,07	0	5,19	39,15
14	ogr14	3,78	4,08	3,99	3,1	4,87	4,47	3,65	0,66	4,24	3,75	36,59
15	ogr15	6,76	9,85	4,72	4,73	7,96	1,29	4,33	4,41	9,5	3,25	56,8
16	ogr16	3,78	4,08	3,74	2,96	8,25	5,16	4,33	0	5,72	3,18	41,2
17	ogr17	2,67	10	3,68	7,04	5,85	8,94	4,63	3,28	1,41	4	51,5
18	ogr18	5,67	4,08	3,93	1,96	1,27	6,32	0	3,94	4,9	4,38	36,45
19	ogr19	3,78	9,67	3,46	8,85	8,25	3,16	5,81	4,15	6,32	1,79	55,24
20	ogr20	3,09	4,08	2,95	8,85	8,25	4,47	0	0	2	0,78	34,47
21	ogr21	3,09	4,08	3,96	3,24	4,47	5,16	0	0,93	1,41	4,6	30,94
22	ogr22	2,18	10	2,53	2,94	9,84	6,32	0	1,77	5,67	5,83	47,08
23	ogr23	3,78	10	1,65	4,37	8,25	6,32	0	10	4,95	4,08	53,4
24	ogr24	3,78	9,8	1,34	4,2	8,25	6,32	0	8,97	4,95	3,13	50,74
25	ogr25	3,09	10	3,46	3,14	4,71	5,16	0	0,93	3,33	5,37	39,19
26	ogr26	3,78	10	1,83	8,15	8,25	6,32	0	9,31	4,95	4,08	56,67
27	ogr27	6,76	9,85	4,03	4,4	7,08	8,94	5,42	2,94	9,5	4,08	63
28	ogr28	0	3,27	1,27	1,63	8,2	7,75	0	0	0	4,02	26,14
29	ogr29	0	4,08	2,27	2	4,3	5,16	0	0	0,58	3,77	22,16
30	ogr30	4,36	4,08	2,95	2,26	1,05	6,32	0	0,76	0	3,24	25,02
31	ogr31	6,76	9,67	3,47	3,24	8,25	8,94	0	3,09	2,85	5,3	51,57
32	ogr32	5,67	9,96	1,9	8,85	8,82	8,94	0	0,52	2,18	3,89	50,73
33	ogr33	6,55	4,08	1,27	7,24	6,76	10	0	2,43	1,1	7,41	46,84
34	ogr34	3,38	4,08	2,95	6,53	6,75	10	0	1,86	0	6,18	41,73
35	ogr35	3,78	9,47	4,1	8,85	8,25	5,16	4,33	2,11	2,85	4,5	53,4
36	ogr36	2,67	3,27	4,39	2,69	6,69	8,94	0	0	1,41	1,5	31,56
37	ogr37	6,55	4,08	3,04	8,85	9,78	7,75	2,4	0,5	0	4,71	47,66
38	ogr38	6,55	4,08	3,15	8,85	9,84	7,75	0	0,42	0	4,46	45,1
39	ogr39	3,78	3,27	2,53	3,33	5,96	7,75	0	0	1,9	3,13	31,65
40	ogr40	3,78	9,67	4,63	3,07	2,11	6,32	0	2,4	3,21	3,7	38,89
41	ogr41	5,35	3,65	2,53	7,67	2,82	5,16	0	2,25	1,79	4,43	35,65

Birinci Cevap Anahtarına Göre GAA												
Sıra No	Ad Soyad	S1P	S2P	S3P	S4P	S5P	S6P	S7P	S8P	S9P	S10P	Toplam Puan
1	ogr1	6,4	0,23	9,95	0,53	8,69	2,89	9,84	1,43	3,22	0,57	43,75
2	ogr2	7,11	0,54	6,97	9,99	3,42	6,21	9,88	0,89	9,99	8,29	63,29
3	ogr3	8,1	0,54	9,98	8,11	1,99	9,26	2,87	1,03	9,85	2,39	54,12
4	ogr4	9,48	0,54	9,57	4,01	2,79	9,8	10	4,56	9,94	4,85	65,54
5	ogr5	9,55	10	3,57	1,25	3,69	9,45	2,87	3,68	9,95	9,32	63,33
6	ogr6	8,46	10	9,89	8,65	2,58	9,8	9,98	2,19	10	7,05	78,6
7	ogr7	2,03	10	5,47	2,76	1,87	9,96	6,61	5,59	3,98	9,18	57,45
8	ogr8	1,2	10	9,79	3,32	3,35	8,98	9,88	0,31	5,86	9,82	62,51
9	ogr9	2,43	0,54	4,22	1,62	9,39	9,35	3,01	9,83	9,92	9,86	60,17
10	ogr10	8,14	10	4,25	8,98	2,51	9,48	9,99	2,63	9,21	9,02	74,21
11	ogr11	9,91	0,54	9,9	6,25	5,71	9,44	3,71	2,48	10	9,62	67,56
12	ogr12	9,02	0,54	9,45	6,21	2,81	9,52	3,71	0,57	10	10	61,83
13	ogr13	9,55	0,54	9,99	2,09	5,37	9,55	9,96	8,24	1,43	6,51	63,23
14	ogr14	9,9	0,54	9,99	1,25	6,02	9,97	6,56	0,1	9,94	9,15	63,42
15	ogr15	9,53	10	9,77	7,01	0,99	3,26	10	6,66	10	8,43	75,65
16	ogr16	9,48	0,54	9,99	3,04	1,9	9,35	10	4,04	9,99	8,72	67,05
17	ogr17	2,81	10	9,91	9,52	0,45	9,47	8,08	4,34	9,22	9,97	73,77
18	ogr18	7,46	0,54	9,91	0,58	4,25	9,54	2,3	8,41	10	9,61	62,6
19	ogr19	9,48	9,84	9,78	10	1,9	9,8	6,6	8,23	10	9,98	85,61
20	ogr20	3,37	0,54	2,29	10	1,9	9,2	2,87	2,94	7,61	8,65	49,37
21	ogr21	3,59	0,54	6,07	7,04	6,3	9,58	3,01	1,27	9,61	9,19	56,2
22	ogr22	4,06	10	4,89	1,47	9,98	9,4	3,01	3,35	9,96	9,98	66,1
23	ogr23	7,02	10	3,57	0	1,9	9,4	9,99	10	10	9,12	71
24	ogr24	7,02	10	0,81	0,12	1,9	9,4	9,99	10	10	8,93	68,17
25	ogr25	3,59	9,91	9,78	5,81	0,49	9,58	3,01	1,27	9,27	9,14	61,85
26	ogr26	7,02	10	4,72	9,62	1,9	9,54	9,91	10	10	9,12	81,83
27	ogr27	9,35	10	9,69	7,79	9,97	9,43	7,45	6,58	10	8,62	88,88
28	ogr28	8,43	0,47	9,51	4,64	3,27	9,33	0,96	2,64	7,08	4,78	51,11
29	ogr29	9,55	0,54	5,33	0,11	1,15	9,14	3,23	2,34	9,67	9,85	50,91
30	ogr30	4,71	0,54	9,8	9,53	4,4	9,16	3,64	1,73	0,58	9,61	53,7
31	ogr31	9,98	9,55	1,74	0,82	1,9	9,43	3,73	7,82	0,48	9,45	54,9
32	ogr32	6,57	9,79	0,04	10	6,6	9,43	2,33	3,94	5,02	9,61	63,33
33	ogr33	3,93	0,54	9,51	9,92	0,19	9,41	9,95	2,12	7,82	9,83	63,22
34	ogr34	3,24	0,54	2,29	9,25	1,2	9,37	9,95	1,69	9,81	9,98	57,32
35	ogr35	8,99	0,02	0,64	10	1,9	9,57	10	4,02	0,48	0,11	45,73
36	ogr36	9,67	0,46	10	0,37	1,03	9,39	0,96	3,67	9,32	9,65	54,52
37	ogr37	7,69	0,54	6,57	10	9,97	9,42	10	3,87	2,92	9,73	70,71
38	ogr38	7,69	0,54	9,78	10	9,98	9,42	9,99	3,82	2,92	8,87	73,01
39	ogr39	9,66	0,46	3,69	0,92	1,62	9,45	2,16	1,82	10	8,51	48,29
40	ogr40	2,67	9,57	8,09	9,25	1,33	9,54	2,87	6,77	9,13	9,86	69,08
41	ogr41	2,75	0,53	4,89	9,91	4,27	9,57	0,96	1,31	9,8	9,9	53,89

Birinci Cevap Anahtarına Göre TGAA												
Sıra No	Ad Soyad	S1P	S2P	S3P	S4P	S5P	S6P	S7P	S8P	S9P	S10P	Toplam Puan
1	ogr1	4,26	0,5	9,86	1,28	6,61	0,42	9,94	2,75	3,25	2,29	41,16
2	ogr2	6,16	0,5	6,91	9,77	3,91	2,71	9,93	1,69	9,99	8,15	59,72
3	ogr3	9,68	0,5	9,8	8,29	2,02	5,02	2,77	2,28	9,84	2,2	52,4
4	ogr4	9,99	0,5	9,55	3,81	2,73	7,18	10	5,15	9,97	6,03	64,91
5	ogr5	10	9,86	2,89	0,77	3,84	9,98	2,77	1,44	9,96	9,47	60,98
6	ogr6	9,85	10	9,89	8,45	2,69	7,18	9,99	2,01	10	7,78	77,84
7	ogr7	0,74	10	4,68	2,51	2,03	9,94	6,64	6,62	3,27	9,25	55,68
8	ogr8	1,69	10	9,41	3,03	3,61	8,67	9,93	0,45	7,11	9,88	63,78
9	ogr9	0,8	0,5	3,12	1,24	9,41	9,76	2,96	9,95	9,91	9,74	57,39
10	ogr10	9,7	10	5,63	9,91	2,59	9,85	10	0,66	9,27	9,35	76,96
11	ogr11	9,82	0,5	9,95	6,33	6,77	9,97	3,64	4,16	10	9,3	70,44
12	ogr12	9,95	0,5	9,52	7,83	3,81	10	3,64	2,27	10	9,99	67,51
13	ogr13	9,68	0,5	10	4,53	6,25	9,97	9,97	3,99	1,27	6,58	62,74
14	ogr14	9,94	0,5	9,97	1,57	6,86	9,11	6,68	2,45	9,96	9,08	66,12
15	ogr15	9,41	9,83	9,59	8,01	1,06	4,29	10	7,61	10	8,1	77,9
16	ogr16	9,98	0,5	10	3,92	2,03	9,87	10	4,35	10	8,62	69,27
17	ogr17	1,11	10	9,96	9,3	0,69	9,99	8,15	5,47	9,1	9,99	73,76
18	ogr18	6,65	0,5	9,85	0,32	4,56	9,23	2,14	7,14	10	9,89	60,28
19	ogr19	9,98	9,83	9,77	9,98	2,03	7,18	6,59	7,13	10	9,96	82,45
20	ogr20	0,88	0,5	1,99	9,98	2,03	9,62	2,77	4,47	7,6	8,06	47,9
21	ogr21	1,11	0,5	9,52	7,48	7,26	9,97	2,96	0,15	9,61	8,89	57,45
22	ogr22	2,21	10	3,1	1,09	9,99	9,64	2,96	1,68	9,95	10	60,62
23	ogr23	9,09	10	2,89	1,63	2,03	9,64	10	10	9,99	9,25	74,52
24	ogr24	9,09	10	2,73	1,88	2,03	9,64	10	10	9,99	8,96	74,32
25	ogr25	1,11	10	9,77	6,03	1,39	9,97	2,96	0,15	9,65	8,53	59,56
26	ogr26	9,09	10	5,85	9,82	2,03	9,23	9,94	10	9,99	9,25	85,2
27	ogr27	10	9,86	9,76	7,53	9,99	10	7,44	6,84	10	8,23	89,65
28	ogr28	6,34	0,61	8,66	4,94	3,77	9,98	0,69	4,22	7,88	4,93	52,02
29	ogr29	7,93	0,5	4,62	0,07	1,15	9,64	3,14	3,97	9,64	9,61	50,27
30	ogr30	4,9	0,5	9,94	8,41	4,73	9,64	3,66	1,68	1,03	9,39	53,88
31	ogr31	9,88	9,88	1,17	0,9	2,03	10	3,74	8,48	0,52	9,66	56,26
32	ogr32	8,49	9,76	0,95	10	6,56	10	2,39	4,95	4,9	9,38	67,38
33	ogr33	6,67	0,5	8,66	9,98	0,57	10	9,98	0,89	8,3	9,73	65,28
34	ogr34	0,74	0,5	1,99	9,91	1,46	10	9,98	0,38	9,77	10	54,73
35	ogr35	9,96	8,01	0,81	9,98	2,03	9,64	10	4,67	0,52	0,63	56,25
36	ogr36	8,54	0,57	9,98	0,25	1,08	9,98	0,69	4,42	9,33	8,86	53,7
37	ogr37	9,43	0,5	5,55	9,98	9,99	9,94	10	4,74	2,89	9,88	72,9
38	ogr38	9,43	0,5	9,84	9,98	9,99	9,94	9,99	4,72	2,89	9,36	76,64
39	ogr39	8,46	0,57	3,1	1	1,88	9,76	2,53	3,63	10	9,2	50,13
40	ogr40	3,78	9,86	8,29	8,18	2,16	9,23	2,77	4,94	9,34	9,69	68,24
41	ogr41	6,16	0,55	3,1	9,91	4,47	9,64	0,69	0,72	9,8	9,9	54,94

EK N. İkinci Cevap Anahtarına Göre Kosinüs Benzerlik, GAA ve TGAA Puanları

İkinci Cevap Anahtarına Göre Kosinüs Benzerliği												
Sıra No	Ad Soyad	S1P	S2P	S3P	S4P	S5P	S6P	S7P	S8P	S9P	S10P	Toplam Puan
1	ogr1	0	7,07	2,47	2,41	2,37	0,83	0	0,44	0,63	2,33	18,55
2	ogr2	5,35	7,3	3,25	4,55	2,05	2,24	0	1,22	5,52	4,74	36,22
3	ogr3	5,07	7,3	1,8	1,91	5,13	3,65	0	1,18	2,11	0	28,15
4	ogr4	8,02	7,3	4,29	3,12	4,59	3,16	4,08	0,96	4,47	3,2	43,19
5	ogr5	6,17	7,11	1,35	1,18	9,46	9,49	0	2,13	3,07	2,24	42,2
6	ogr6	5,35	6,71	4,5	4,49	9,23	3,16	3,16	0	8,06	2,98	47,64
7	ogr7	3,78	6,71	2,02	3,16	5,13	7,75	3,33	0	0	4,33	36,21
8	ogr8	4,36	6,39	1,35	0	3,82	2,24	0	0,59	1,83	2,5	23,08
9	ogr9	2,86	7,3	1,8	1,83	4,5	4,47	0	2,11	3,87	6,24	34,98
10	ogr10	4,63	6,71	3,6	5,52	5,62	5,16	0,96	0,43	1,2	2,04	35,87
11	ogr11	7,56	7,3	5,78	4,56	8,9	8,94	0	0	8,22	5,82	57,08
12	ogr12	7,72	7,3	4,66	2,66	7,89	9,13	0	0	6,59	5,77	51,72
13	ogr13	5,67	7,3	5,81	1,67	7,36	4,47	0	1,75	0	4,44	38,47
14	ogr14	3,78	7,3	4,25	2,79	5,3	4,47	3,16	0,83	3,35	4,42	39,65
15	ogr15	8,45	7,11	4,74	4,33	9,37	1,29	7,5	3,87	7,91	3,56	58,13
16	ogr16	5,67	7,3	4,16	2,92	9,75	5,16	7,5	0	5,16	2,5	50,12
17	ogr17	2,67	6,71	3,32	5,43	7,64	8,94	6,68	4,17	2,24	4,12	51,92
18	ogr18	5,67	7,3	4,22	1,66	1,38	6,32	0	3,33	3,87	4,3	38,05
19	ogr19	5,67	5,96	2,94	8,34	9,75	3,16	3,35	4,22	4	2,11	49,5
20	ogr20	3,09	7,3	3,02	8,34	9,75	4,47	0	0	3,16	0,92	40,05
21	ogr21	3,09	7,3	4,06	3,12	4,87	5,16	0	1,18	2,24	4,77	35,79
22	ogr22	2,18	6,71	1,95	2,87	8,09	6,32	0	1,92	2,76	5,66	38,46
23	ogr23	5,67	6,71	1,35	4,17	9,75	6,32	0	8,32	4,47	4,33	51,09
24	ogr24	5,67	6,39	1,1	4,06	9,75	6,32	0	7,44	4,47	2,11	47,31
25	ogr25	3,09	6,71	2,94	2,95	6,16	5,16	0	1,18	2,63	4	34,82
26	ogr26	5,67	6,71	1,77	8	9,75	6,32	0	7,44	4,47	4,33	54,46
27	ogr27	6,76	7,11	2,78	4,06	6,52	8,94	5,21	2,24	7,91	4,33	55,86
28	ogr28	0	5,48	1,56	2,21	9,73	7,75	0	0	0	3,69	30,42
29	ogr29	0	7,3	0,7	2,25	5,62	5,16	0	0	0,91	3,89	25,83
30	ogr30	4,36	7,3	2,41	2,55	2,29	6,32	2,04	0,96	0	3,44	31,67
31	ogr31	6,76	7,45	3,68	2,99	9,75	8,94	1,89	3,27	1,8	5,21	51,74
32	ogr32	5,67	7,13	2,34	8,23	8,9	8,94	2,5	0	1,38	4,17	49,26
33	ogr33	6,55	7,3	1,56	5,59	7,97	10	0	2,57	1,73	8	51,27
34	ogr34	3,38	7,3	3,02	6,13	8,72	10	0	1,77	0	5,96	46,28
35	ogr35	5,67	8,72	3,62	8,34	9,75	5,16	2,5	1,91	1,8	4,12	51,59
36	ogr36	5,35	7,3	3,71	2,6	8,61	8,94	0	0	2,24	1,77	40,52
37	ogr37	6,55	7,3	2,87	8,34	8,05	7,75	2,77	0	0	4,58	48,21
38	ogr38	6,55	7,3	2,66	8,34	8,09	7,75	3,54	0	0	4,33	48,56
39	ogr39	3,78	7,3	1,95	3,1	7,3	7,75	0	0	3	3,16	37,34
40	ogr40	3,78	5,96	4,22	3,46	2,29	6,32	0	0,61	2,54	4	33,18
41	ogr41	5,35	6,12	1,95	6,91	3,68	5,16	0	2,29	2,83	3,73	38,02

İkinci Cevap Anahtarına Göre GAA												
Sıra No	Ad Soyad	S1P	S2P	S3P	S4P	S5P	S6P	S7P	S8P	S9P	S10P	Toplam Puan
1	ogr1	3,61	4,71	9,89	0,07	6,88	0,81	9,89	1,29	2,81	0,59	40,55
2	ogr2	8,99	4,44	6,59	9,97	8,31	2,91	9,84	0,75	9,96	8,88	70,64
3	ogr3	9,55	4,44	10	7,82	9,04	7,23	2,6	1,16	9,77	3,49	65,1
4	ogr4	10	4,44	9,71	3,58	8,66	9,85	9,99	4,69	9,88	3,81	74,61
5	ogr5	10	8,66	4,04	0,78	9,91	9,99	2,6	3,81	9,9	9,68	69,37
6	ogr6	9,72	8,71	9,95	8,4	10	9,85	10	2,33	9,99	6,19	85,14
7	ogr7	1,21	8,71	5,04	2,31	9,09	9,59	6,82	5,47	4,38	9,58	62,2
8	ogr8	2,04	8,68	9,67	3,75	8,35	6,74	9,84	0,18	5,5	9,97	64,72
9	ogr9	0,8	4,44	3,76	2,08	1,11	10	2,75	9,85	9,86	9,99	54,64
10	ogr10	9,57	8,71	3,78	8,77	8,8	9,99	9,98	2,5	9,03	9,46	80,59
11	ogr11	9,82	4,44	9,82	5,88	9,33	9,99	3,45	2,34	9,99	9,87	74,93
12	ogr12	9,93	4,44	9,28	5,84	9,99	9,98	3,45	0,43	9,98	9,9	73,22
13	ogr13	8,1	4,44	9,96	1,63	9,47	9,97	9,93	8,32	1,86	5,59	69,27
14	ogr14	9,83	4,44	10	0,78	9,19	8,97	6,77	0,04	9,88	9,56	69,46
15	ogr15	8,06	8,65	9,65	6,67	9,9	0,43	10	6,77	10	8,99	79,12
16	ogr16	7,97	4,44	10	2,59	9,99	10	10	3,91	9,95	9,22	78,07
17	ogr17	0,41	8,71	9,96	9,37	9,59	9,99	8,24	4,46	9,04	9,82	79,59
18	ogr18	9,2	4,44	9,96	0,11	7,78	9,98	2,02	8,48	9,98	9,86	71,81
19	ogr19	7,97	9,45	9,87	10	9,99	9,85	6,81	8,31	9,98	9,84	92,07
20	ogr20	0,19	4,43	1,79	10	9,99	10	2,6	2,81	7,31	9,17	58,29
21	ogr21	0,41	4,44	5,66	6,7	9,05	9,96	2,75	1,41	9,48	8,67	58,53
22	ogr22	0,92	8,71	4,43	1,93	1,72	10	2,75	3,22	9,91	9,84	53,43
23	ogr23	8,93	8,71	4,04	0,47	9,99	10	10	10	9,97	9,54	81,65
24	ogr24	8,93	8,79	0,3	0,59	9,99	10	10	10	9,97	9,39	77,96
25	ogr25	0,41	9,28	9,87	5,43	9,82	9,96	2,75	1,41	9,1	8,61	66,64
26	ogr26	8,93	8,71	4,27	9,48	9,99	9,98	9,87	10	9,97	9,54	90,74
27	ogr27	9,99	8,66	9,81	7,49	1,57	9,99	7,63	6,69	9,99	9,14	80,96
28	ogr28	6,27	4,5	9,34	4,22	9,96	10	0,68	2,5	6,76	3,74	57,97
29	ogr29	8,1	4,43	5,75	0,36	9,92	9,99	2,96	2,2	9,55	9,98	63,24
30	ogr30	7,29	4,44	9,89	9,38	7,67	9,99	3,38	1,86	0,14	9,87	63,91
31	ogr31	9,65	9,78	1,24	1,28	9,99	9,99	3,47	7,9	0,92	9,01	63,23
32	ogr32	8,64	7,53	0,55	9,99	8,87	9,99	2,06	3,81	4,64	9,86	65,94
33	ogr33	6,66	4,44	9,34	9,85	9,75	10	9,98	2,26	7,53	9,98	79,79
34	ogr34	0,04	4,43	1,79	9,07	9,36	10	9,98	1,83	9,88	9,84	66,22
35	ogr35	9,92	4,93	1,15	10	9,99	9,97	9,99	4,14	0,92	1,27	62,28
36	ogr36	8,35	4,5	9,99	0,09	9,41	10	0,68	3,54	9,15	9,89	65,6
37	ogr37	9,33	4,44	6,18	10	1,65	10	10	3,74	3,34	9,39	68,07
38	ogr38	9,33	4,44	9,87	10	1,72	10	10	3,69	3,34	8,28	70,67
39	ogr39	8,32	4,5	3,21	1,38	9,2	9,99	1,88	1,68	9,99	7,84	57,99
40	ogr40	5,61	9,76	7,78	9,06	9,31	9,98	2,6	6,87	8,95	9,99	79,91
41	ogr41	5,68	4,44	4,43	9,96	7,77	9,97	0,68	1,45	9,71	9,66	63,75

İkinci Cevap Anahtarına Göre TGAA												
Sıra No	Ad Soyad	S1P	S2P	S3P	S4P	S5P	S6P	S7P	S8P	S9P	S10P	Toplam Puan
1	ogr1	4,26	9,41	9,7	2,64	9,07	0,42	9,96	2,66	2,79	1,28	52,19
2	ogr2	4,23	9,41	6,61	9,71	7,71	2,71	9,9	1,01	9,95	8,71	69,95
3	ogr3	8,59	9,41	9,92	8,68	8,82	5,02	2,55	1,99	9,74	3,21	67,93
4	ogr4	9,97	9,41	9,73	5,01	8,45	7,18	9,99	5,19	9,91	5,19	80,03
5	ogr5	9,99	4,63	3,43	2,3	9,94	9,98	2,55	2,28	9,91	9,78	64,79
6	ogr6	9,99	3,03	9,97	7,99	10	7,18	10	2,22	9,99	7,11	77,48
7	ogr7	0,08	3,03	4,18	3,62	8,81	9,94	6,81	6,23	3,73	9,6	56,03
8	ogr8	0,87	3,08	9,15	2,45	7,91	8,67	9,9	0,32	6,77	9,99	59,11
9	ogr9	9,42	9,41	2,57	0,76	0,57	9,76	2,74	9,98	9,84	9,92	64,97
10	ogr10	8,61	3,03	5,05	9,96	8,52	9,85	9,99	2,08	9,07	9,67	75,83
11	ogr11	9,93	9,41	9,85	7,03	8,98	9,97	3,43	3,57	9,99	9,63	81,79
12	ogr12	9,94	9,41	9,27	8,61	9,95	10	3,43	1,63	9,98	9,98	82,2
13	ogr13	9,66	9,41	9,97	5,45	9,26	9,97	9,95	4,49	1,74	5,83	75,73
14	ogr14	9,89	9,41	10	1,53	8,93	9,11	6,85	1,81	9,91	9,47	76,91
15	ogr15	9,45	4,76	9,47	8,47	9,83	4,29	10	7,96	9,99	8,67	82,89
16	ogr16	9,99	9,41	9,99	5,14	9,96	9,87	10	4,07	9,97	9,1	87,5
17	ogr17	1,55	3,03	9,99	9,5	9,37	9,99	8,28	5,82	8,89	9,9	76,32
18	ogr18	5,32	9,41	9,95	1,07	7,23	9,23	1,92	7,57	9,99	10	71,69
19	ogr19	9,99	1,19	9,9	9,95	9,96	7,18	6,76	7,43	9,98	9,82	82,16
20	ogr20	1,16	9,41	1,46	9,95	9,96	9,62	2,55	3,85	7,28	8,62	63,86
21	ogr21	1,38	9,41	9,26	7,78	8,66	9,97	2,74	0,89	9,47	8,4	67,96
22	ogr22	2,38	3,03	2,53	0,78	2,39	9,64	2,74	1,23	9,89	9,96	44,57
23	ogr23	9,24	3,03	3,43	3,04	9,96	9,64	10	9,99	9,95	9,6	77,88
24	ogr24	9,24	2,88	2,1	3,37	9,96	9,64	10	9,98	9,95	9,39	76,51
25	ogr25	1,38	1,76	9,9	6,47	9,89	9,97	2,74	0,89	9,51	7,77	60,28
26	ogr26	9,24	3,03	5,44	9,93	9,96	9,23	9,92	9,99	9,95	9,6	86,29
27	ogr27	9,63	4,63	9,89	8,24	2,35	10	7,59	7,23	9,98	8,78	78,32
28	ogr28	6,45	9,37	8,3	1,19	9,95	9,98	0,47	3,59	7,58	4,01	60,89
29	ogr29	8,39	9,41	5,17	0,23	9,85	9,64	2,92	3,35	9,5	9,84	68,3
30	ogr30	2,42	9,41	9,93	9,3	7,1	9,64	3,45	0,94	0,55	9,69	62,43
31	ogr31	9,06	1,67	0,59	0,65	9,96	10	3,53	8,65	1	9,35	54,46
32	ogr32	6,41	5,05	1,51	9,99	9,1	10	2,17	4,43	4,47	9,69	62,82
33	ogr33	4,6	9,41	8,3	9,98	9,41	10	9,99	1,6	8,02	9,92	81,23
34	ogr34	1,02	9,41	1,46	9,94	9,07	10	9,99	1,07	9,86	9,92	71,74
35	ogr35	10	8,21	1,49	9,95	9,96	9,64	9,99	4,96	1	0,41	65,61
36	ogr36	8,54	9,39	10	1,13	9,22	9,98	0,47	3,95	9,15	9,29	71,12
37	ogr37	8,26	9,41	5,45	9,95	2,32	9,94	10	4,25	3,35	9,68	72,61
38	ogr38	8,26	9,41	9,89	9,95	2,39	9,94	10	4,23	3,35	8,95	76,37
39	ogr39	8,41	9,39	2,53	0,76	8,88	9,76	2,31	2,98	9,98	8,75	63,75
40	ogr40	1,74	0,08	7,74	9,14	8,74	9,23	2,55	5,35	9,16	9,89	63,62
41	ogr41	4,23	9,4	2,53	9,87	7,3	9,64	0,47	1,42	9,69	10	64,55

EK O. Eđitmen1 Cevap Anahtarı

1 OSI referans modelinin birinci katmanındaki cihazları yazınız

NIC, Yineleyici, Hub, MAU, Kablo, Alıcı ve verici

2 Ađ topoloji türlerini yazınız

Dođrusal Halka Yıldız Ađaç Karmaşık Topoloji

3 OSI referans modelindeki veri iletim katmanının özelliklerini yazınız
Ađ katmanından aldığı veri paketlerine hata kontrol bitlerini ekleyerek çerçeve (frame) halinde fiziksel katmana iletme işinden sorumludur. İletilen çerçevenin doğru mu yanlış mı iletildiđini kontrol eder, eđer çerçeve hatalı iletilmişse çerçevenin yeniden gönderilmesini sağlar. Ayrıca ađ üzerindeki diđer bilgisayarları tanımlama, kablonun o anda kimin tarafından kullanıldığının tespitini yapar. Örn: Ethernet, Frame Relay, ISDN, Switch ve Bridge

4 Dođrusal Topolojinin dezavantajlarını yazınız

Omurga kabloda bir bozulma veya kesilme olursa tüm ađ bağlantısı kesilir. Kablonun sonunda sonlandırıcı (Terminator) olmalıdır. Ađda sorun olduđunda sorunun nerden kaynaklandığının bulmak zaman alıcı olabilir. Tek başına tüm bir binanın ađ çözümü için genellikle kullanılmamaktadır. Çarpışma

5 Topoloji nedir?

Bir ađdaki bilgisayarların nasıl yerleşeceğini, nasıl bağlanacağını, veri iletiminin nasıl olacağını belirleyen genel yapıdır. Fiziksel topoloji: Ađın fiziksel olarak nasıl görüneceđini belirler (Fiziksel katman) Mantıksal topoloji: Bir ađdaki veri akışının nasıl olacağını belirler (Veri iletim katmanı)

6 Ađda kullanılan veri adlarını yazınız

Data segment packet frame bits

7 Fiber optik çeşitlerini yazınız

Simplex fiber Duplex fiber Multi fiber

8 Yineleyici nedir?

Ağ kablosunun erişebileceği maksimum mesafeyi uzatırlar. Ağdaki maksimum düğüm sayısını arttırır. Kablo arızalarının etkisini azaltabilir. Farklı kablo tipleri kullanan ağları birleştirebilir.

9 Ağın avantajları?

Verilerin paylaşımı Dosyaların paylaşımı Donanımların paylaşımı Yazılımların paylaşımı Yönetim ve Güvenlik Erişim ve İletişim Eğitim

10 OSI referans modelindeki ağ (network) katmanının özelliklerini yazınız

Bu katmanda iletilen veri blokları paket olarak adlandırılır. Bu katman, veri paketlerinin ağ adreslerini kullanarak bu paketleri uygun ağlara yönlendirme işini yapar. Adresleme işlemlerini (Mantıksal adres ve fiziksel adres çevrimleri) yürütür. Yönlendiriciler (Router) bu katmanda tanımlıdır. Örnek; IP ve IPX.

EK Ö. Eğitim2 Cevap Anahtarı

1 OSI referans modelinin birinci katmanındaki cihazları yazınız

Ethernet, Repeater, Hub, Multi Access Unit, Kablo, Alıcı ve verici

2 Ağ topoloji türlerini yazınız

Bus Ring Star Tree Mesh Topoloji

3 OSI referans modelindeki veri iletim katmanının özelliklerini yazınız

İletilen çerçevenin doğru mu yoksa yanlış mı iletildiğini kontrol eder, eğer çerçeve hatalı iletilmişse çerçevenin yeniden gönderilmesini sağlar. Ağ katmanından aldığı veri paketlerine hata kontrol bitlerini ekleyerek çerçeve (frame) halinde fiziksel katmana iletme işinden sorumludur.

4 Doğrusal Topolojinin dezavantajlarını yazınız

Sonlandırıcı kullanmak zorunludur. Omurga kabloda bir bozulma veya kesilme olursa tüm ağ bağlantısı kesilir. Çarpışma çoktur. Ağda sorun olduğunda sorunun nerden kaynaklandığını bulmak zaman alıcı olabilir. Tek başına tüm bir binanın ağ çözümü için genellikle kullanılmamaktadır.

5 Topoloji nedir?

Bir ağdaki bilgisayarların nasıl yerleşeceğini, nasıl bağlanacağını, veri iletiminin nasıl olacağını belirleyen genel yapıdır.

6 Ağda kullanılan veri adlarını yazınız

Veri bölüm paket çerçeve bit

7 Fiber optik çeşitlerini yazınız

Tekli fiber ikili fiber çoklu fiber

8 Yineleyici nedir?

Ağın maksimum mesafesini uzatır. Ağdaki bilgisayar sayısını artırır. Kablo arızalarını azaltır ve farklı kablo tiplerindeki ağları birleştirebilir.

9 Ađın avantajları?

Verilerin, Dosyaların, Donanımların, Yazılımların paylaşımı, Yönetim, Güvenlik Erişim, İletişim, Eğitim

100Sİ referans modelindeki ađ (network) katmanının özelliklerini yazınız

Bu katman, veri paketlerinin ađ adreslerini kullanarak bu paketleri uygun ađlara yönlendirme işini yapar. Adresleme işlemlerini yürütür. Bu katmanda iletilen veri blokları paket olarak adlandırılır. Yönlendiriciler (Router), IP ve IPX.



EK P. TSPS Puan ile Öğretmen1 Puan Pearson t Testi Sonuçları

		Levene's Test for Equality of Variances		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
		Lower	Upper	Lower	Upper	Lower	Upper	Lower	Upper	
soru 1	Equal variances assumed	,041	,839	-,862	80	,391	-,36561	,42393	-1,20926	,47804
	Equal variances not assumed			-,862	79,969	,391	-,36561	,42393	-1,20926	,47804
soru 2	Equal variances assumed	15,334	,000	,684	80	,496	,42439	,62042	-,81029	1,65907
	Equal variances not assumed			,684	77,818	,496	,42439	,62042	-,81082	1,65960
soru 3	Equal variances assumed	1,579	,213	-,882	80	,381	-,62488	,70882	-2,03547	,78571
	Equal variances not assumed			-,882	79,255	,381	-,62488	,70882	-2,03567	,78592
soru 4	Equal variances assumed	2,178	,144	,317	80	,752	,24927	,78558	-1,31409	1,81262
	Equal variances not assumed			,317	78,875	,752	,24927	,78558	-1,31443	1,81297
soru 5	Equal variances assumed	,059	,809	,137	80	,891	,07610	,55452	-1,02743	1,17963
	Equal variances not assumed			,137	79,898	,891	,07610	,55452	-1,02746	1,17965
soru 6	Equal variances assumed	,047	,830	-,220	80	,826	-,12146	,55147	-1,21893	,97601
	Equal variances not assumed			-,220	79,945	,826	-,12146	,55147	-1,21894	,97602
soru 7	Equal variances assumed	,042	,838	-,094	80	,925	-,04293	,45579	-,94998	,86413
	Equal variances not assumed			-,094	79,949	,925	-,04293	,45579	-,94999	,86414
soru 8	Equal variances assumed	,012	,914	-,463	80	,644	-,25195	,54385	-1,33424	,83034
	Equal variances not assumed			-,463	79,999	,644	-,25195	,54385	-1,33424	,83034
soru 9	Equal variances assumed	,094	,760	-,173	80	,863	-,11854	,68501	-1,48176	1,24468
	Equal variances not assumed			-,173	79,918	,863	-,11854	,68501	-1,48178	1,24470
soru 10	Equal variances assumed	,436	,511	-,671	80	,504	-,34244	,51061	-1,35858	,67370
	Equal variances not assumed			-,671	79,520	,504	-,34244	,51061	-1,35867	,67379

EK R. TSPS Puan ile Öğretmen2 Puan Pearson t Testi Sonuçları

		Levene's Test for Equality of Variances		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
		Lower	Upper	Lower	Upper	Lower	Upper	Lower	Upper	
soru 1	Equal variances assumed	,041	,839	-,862	80	,391	-,36561	,42393	-1,20926	,47804
	Equal variances not assumed			-,862	79,969	,391	-,36561	,42393	-1,20926	,47804
soru 2	Equal variances assumed	15,334	,000	,684	80	,496	,42439	,62042	-,81029	1,65907
	Equal variances not assumed			,684	77,818	,496	,42439	,62042	-,81082	1,65960
soru 3	Equal variances assumed	1,733	,192	-,698	80	,487	-,50244	,72021	-1,93569	,93082
	Equal variances not assumed			-,698	79,103	,487	-,50244	,72021	-1,93594	,93107
soru 4	Equal variances assumed	1,669	,200	-,046	80	,964	-,03561	,77803	-1,58393	1,51271
	Equal variances not assumed			-,046	79,172	,964	-,03561	,77803	-1,58418	1,51296
soru 5	Equal variances assumed	,007	,936	,015	80	,988	,00902	,58603	-1,15721	1,17526
	Equal variances not assumed			,015	79,981	,988	,00902	,58603	-1,15721	1,17526
soru 6	Equal variances assumed	,047	,830	-,220	80	,826	-,12146	,55147	-1,21893	,97601
	Equal variances not assumed			-,220	79,945	,826	-,12146	,55147	-1,21894	,97602
soru 7	Equal variances assumed	,042	,838	-,094	80	,925	-,04293	,45579	-,94998	,86413
	Equal variances not assumed			-,094	79,949	,925	-,04293	,45579	-,94999	,86414
soru 8	Equal variances assumed	,013	,910	-,174	80	,862	-,08122	,46554	-1,00768	,84524
	Equal variances not assumed			-,174	79,988	,862	-,08122	,46554	-1,00768	,84524
soru 9	Equal variances assumed	2,494	,118	,322	80	,748	,18439	,57262	-,95517	1,32395
	Equal variances not assumed			,322	77,827	,748	,18439	,57262	-,95565	1,32443
soru 10	Equal variances assumed	,057	,812	-,713	80	,478	-,37488	,52568	-1,42102	,67126
	Equal variances not assumed			-,713	80,000	,478	-,37488	,52568	-1,42102	,67126

EK S. TSPSORT Puan ile Öğretmen1 Puan Pearson t Testi Sonuçları

		Levene's Test for Equality of Variances		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
		Lower	Upper	Lower	Upper	Lower	Upper	Lower	Upper	
soru 1	Equal variances assumed	,041	,839	-,862	80	,391	-,36561	,42393	-1,20926	,47804
	Equal variances not assumed			-,862	79,969	,391	-,36561	,42393	-1,20926	,47804
soru 2	Equal variances assumed	15,334	,000	,684	80	,496	,42439	,62042	-,81029	1,65907
	Equal variances not assumed			,684	77,818	,496	,42439	,62042	-,81082	1,65960
soru 3	Equal variances assumed	1,874	,175	-,791	80	,431	-,56366	,71225	-1,98108	,85376
	Equal variances not assumed			-,791	79,117	,431	-,56366	,71225	-1,98132	,85400
soru 4	Equal variances assumed	2,010	,160	,137	80	,891	,10683	,77883	-1,44309	1,65675
	Equal variances not assumed			,137	79,142	,891	,10683	,77883	-1,44335	1,65701
soru 5	Equal variances assumed	,114	,737	-,650	80	,517	-,35988	,55342	-1,46122	,74147
	Equal variances not assumed			-,650	79,918	,517	-,35988	,55342	-1,46124	,74148
soru 6	Equal variances assumed	,047	,830	-,220	80	,826	-,12146	,55147	-1,21893	,97601
	Equal variances not assumed			-,220	79,945	,826	-,12146	,55147	-1,21894	,97602
soru 7	Equal variances assumed	,042	,838	-,094	80	,925	-,04293	,45579	-,94998	,86413
	Equal variances not assumed			-,094	79,949	,925	-,04293	,45579	-,94999	,86414
soru 8	Equal variances assumed	,018	,893	-,201	80	,841	-,10561	,52414	-1,14867	,93745
	Equal variances not assumed			-,201	79,476	,841	-,10561	,52414	-1,14878	,93756
soru 9	Equal variances assumed	,625	,431	,147	80	,884	,09390	,63923	-1,17821	1,36602
	Equal variances not assumed			,147	79,015	,884	,09390	,63923	-1,17846	1,36626
soru 10	Equal variances assumed	,255	,615	-,692	80	,491	-,35866	,51805	-1,38961	,67229
	Equal variances not assumed			-,692	79,824	,491	-,35866	,51805	-1,38964	,67232

EK Ş. TSPSORT Puan ile Öğretmen2 Puan Pearson t Testi Sonuçları

		Levene's Test for Equality of Variances		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
		Lower	Upper	Lower	Upper	Lower	Upper	Lower	Upper	
soru 1	Equal variances assumed	,041	,839	-,862	80	,391	-,36561	,42393	-1,20926	,47804
	Equal variances not assumed			-,862	79,969	,391	-,36561	,42393	-1,20926	,47804
soru 2	Equal variances assumed	15,334	,000	,684	80	,496	,42439	,62042	-,81029	1,65907
	Equal variances not assumed			,684	77,818	,496	,42439	,62042	-,81082	1,65960
soru 3	Equal variances assumed	1,469	,229	-,788	80	,433	-,56366	,71566	-1,98788	,86056
	Equal variances not assumed			-,788	79,283	,433	-,56366	,71566	-1,98807	,86076
soru 4	Equal variances assumed	2,010	,160	,137	80	,891	,10683	,77883	-1,44309	1,65675
	Equal variances not assumed			,137	79,142	,891	,10683	,77883	-1,44335	1,65701
soru 5	Equal variances assumed	,090	,765	,772	80	,442	,44500	,57651	-,70229	1,59229
	Equal variances not assumed			,772	79,808	,442	,44500	,57651	-,70233	1,59233
soru 6	Equal variances assumed	,047	,830	-,220	80	,826	-,12146	,55147	-1,21893	,97601
	Equal variances not assumed			-,220	79,945	,826	-,12146	,55147	-1,21894	,97602
soru 7	Equal variances assumed	,042	,838	-,094	80	,925	-,04293	,45579	-,94998	,86413
	Equal variances not assumed			-,094	79,949	,925	-,04293	,45579	-,94999	,86414
soru 8	Equal variances assumed	,085	,771	-,471	80	,639	-,22756	,48293	-1,18863	,73351
	Equal variances not assumed			-,471	79,462	,639	-,22756	,48293	-1,18873	,73361
soru 9	Equal variances assumed	,164	,686	-,046	80	,963	-,02805	,61060	-1,24319	1,18709
	Equal variances not assumed			-,046	79,944	,963	-,02805	,61060	-1,24320	1,18711
soru 10	Equal variances assumed	,259	,612	-,695	80	,489	-,35866	,51569	-1,38491	,66759
	Equal variances not assumed			-,695	79,884	,489	-,35866	,51569	-1,38493	,66762

ÖZGEÇMİŞ

Adı Soyadı : Ebru YILMAZ İNCE
Doğum Yeri ve Yılı : Finike, 1986
Medeni Hali : Evli
Yabancı Dili : İngilizce
E-posta : ebruince@sdu.edu.tr

Eğitim Durumu

Lise : Isparta Süleyman Demirel Fen Lisesi, 2004
Lisans :SDÜ, Teknik Eğitim Fakültesi, Bilgisayar Sistemleri Öğretmenliği, 2008
Yüksek Lisans :SDÜ, Fen Bilimleri Enstitüsü, Eğitim Teknolojileri, 2011

Mesleki Deneyim

SDÜ Fen Bilimleri Enstitüsü, Arş. Gör. 2008-2010
SDÜ Teknik Bilimler MYO, Öğr. Gör. 2010-..... (halen)

Yayımları

Yılmaz-İnce, E., Demirbilek, M., 2013. Secondary and High School Teachers' Perceptions Regarding Computer Games with Educational Features in Turkey. *Anthropologist*, 16(1-2): 89-96.

Koç, M., Demirbilek, M., Yılmaz-İnce, E., 2015. A needs assessment for academicians' professional development. *Education and Science*, 40(177), 297-311. <http://dx.doi.org/10.15390/EB.2015.2545>

Burs Bilgileri

BİDEB 2211 Tübitak Doktora Bursiyeri

Ulusal kuruluşlarca desteklenen proje yürütücülüğü

SDÜ Grubu, Tübitak 2209/C nolu Üniv. Öğr. Yurtdışı Araş. Pro. Des. Prog.
Meslek Yüksekokullarında Düşünen, Tasarlayan, Geliştiren Beyinler 2012,
Tübitak 2209/B nolu Üniv. Öğr. Yurtdışı Araş. Pro. Des. Prog.

Ulusal kuruluşlarca desteklenen projede görev alma

Tübitak Formula G (Güneş Enerjili Araba Yarışı), SDÜ Solarsonic Takımı Üyesi,
2006.

Proje No: 2408-YL-10, İlk ve Ortaöğretim Öğretmenlerinin Eğitsel Bilgisayar
Oyunları Hakkındaki Görüşleri: Demografik Özelliklere Göre
Karşılaştırma. Süleyman Demirel Üniversitesi Bilimsel Araştırma
Projeleri Yönetim Birimi

Proje No: 2550-M-10, Süleyman Demirel Üniversitesi Öğretim Elemanlarının
Profesyonel Gelişim Eğitimlerine Yönelik İhtiyaç Analizi. Süleyman
Demirel Üniversitesi Bilimsel Araştırma Projeleri Yönetim Birimi

Proje No: 114E952, Öğrenme Yönetim Sistemi Kısa Cevaplı Sorular İçin
Otomatik Değerlendirme Yazılımı: Türkçe Örneği, TÜBİTAK 1002